2011.11

PROGRAMMER 程序员



22 一位消费电子创业者的 Smart TV 创业观

59 技术人员如何转型为 产品经理

64 创业故事 小鸡快跑

68 移动应用 必定不会长久

90 构建开放的软件团队文化

92 在中国应如何改良Scrum框架

102 一脉相承的高可用技术 从分布式系统到流式计算

106 基于Minitransaction 构建可扩展分布式系统

117 移动设备的多核时代

134 纪念 C 语言之父 Dennis M. Ritchie

ISSN 1672-3252

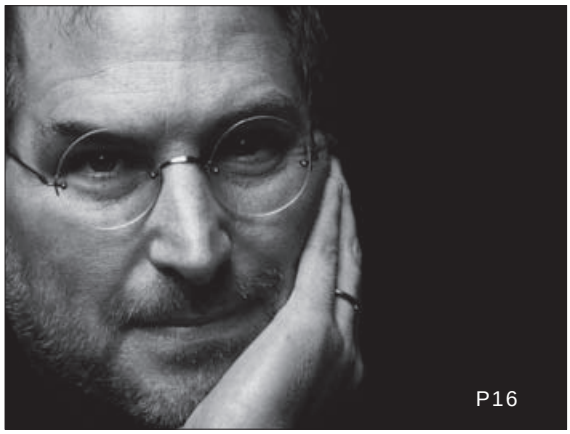
9 771672 325111 11>

邮发代号：2-665 定价：15元

www.programmer.com.cn www.csdn.net

Contents

在本刊文章标题处印有此图标的页面，读者可以使用“一拍”（下载地址为 http://itunes.apple.com/us/app/id463494232?mt=8）拍下整幅文章页面，以阅读该篇文章的更多相关内容。
目前仅支持iPhone/iPad，欢迎使用。





资讯

8 资讯
10 新产品新工具
12 外刊速递
16 程序天下事

热点报道

22 一位消费电子创业者的Smart TV创业观
Android系统已经改变了智能手机市场，会决定和统一未来的智能家电领域吗？让我们与一位资深IT消费电子创业者交流，共同探讨移动平台和智能家电的创业前景。

25 普适计算20年

封面报道
“中国式”产品管理

产品管理诞生于宝洁公司，对于中国企业来说，既老又新，说它老，是因为它的诞生要比现在流行的很多管理思想要早得多；说它新，是因为不但在国内，即使是在世界范围内引起高度关注也是从20世纪80年代末到90年代初才开始的，尤其对于中国企业来说，更是如此。或者可以这样说，在这个管理思想的认识和实践上，我们并没有被国外同行落下太多，我们现在需要做的就是思考如何才能把本来不大的差距继续缩小，直到因地制宜、因时制宜地让产品管理思想本土化、体系化和职业化。我们应该为宝洁审时度势的改造大声叫好，这或许应该成为中国企业实施产品管理的一个标杆。这个标杆建立在宝洁为追求产品管理最合适的实现形式上的不断探索和实践的勇气和耐心中。
中国的产品管理，其实才刚刚上路，我们依然需要在探索中前行。这一切才刚刚开始！

29 中国式产品管理，探索中前行
32 产品经理的“巧实力”：个人素养
35 产品经理的“软实力”：管理知识
38 产品经理的“竞争力”：核心能力
43 互联网产品经理历练之路
48 阿泡的产品管理工具包之产品经理的34个感想
52 打造高效的产品团队
56 产品管理，明天会如何发展？
59 技术人员如何转型为产品经理

Contents



移动

64 《小鸡快跑》的成功故事
一只来自中国的“小鸡”，在亿万应用的竞赛中，跑在前列。笔者深入发掘，为你带来“小鸡”成功背后的精彩故事。

68 移动应用必定不会长久
青蛙设计创意总监Scott Jenson认为移动应用虽然暂时繁华，但“即时”交互的整体概念意味着这种景象并不会长久。

71 从数据角度看手机游戏产业中的佼佼者
移动游戏的平台前景如何？营销推广趋势将有何改变？国内外市场还具备怎样的潜力？谁是目前行业的佼佼者？本文通过数据一一为你解答。

75 用户体验的工作方式

80 用户体验导向的Android应用开发

83 全景解读Android UI自动化测试框架TMTS（下）

管理

87 创业型公司：小公司，大平台
——专访赶集网技术副总裁罗剑
赶集网已为公众所熟知，但对其技术管理者的情况则知之不多。近日，本刊记者专访了赶集网技术副总裁罗剑，请他分享自己的真知灼见。

88 工作就是这么简单
——专访去哪儿网CTO吴永强
早在2007年，吴永强就加入了去哪儿网，全面负责公司的技术管理工作。他将在本文中分享在去哪儿网的难忘经历和技术感悟。

90 构建开放的软件团队文化
伴随着软件开发敏捷化趋势的发展，“以人为本”的软件团队建设理念越来越得到广泛的重视，但在这方面的研究、探讨与实践则显得乏善可陈，现在是到了个体、团队、组织甚至是社会力量一起行动起来的时候了。

92 在中国应如何改良Scrum框架
Scrum在中国得到了越来越多的应用，但仍存在一些缺陷，本文在总结Scrum框架缺陷的基础上，讨论应如何改良Scrum框架，以保证实施成功。

96 毕业选择起步型企业的五大原因

云计算

98 Hypertable：BigTable的高性能实现

102 一脉相承的高可用技术
——从分布式系统到流式计算



2011年11月刊　总第229期

Contents

主管：中国社会科学院
主办：中国社会科学院文献信息中心
出版：《程序员》杂志社
网址：http://www.programmer.com.cn
国际刊号：ISSN 1672-3252
国内刊号：CN11-5038/G2
邮发代号：2-665
广告经营许可证号：京东工商广字0188号

总编：黄长著　Editor-in-chief：Huang Changzhu
社长/常务副总编：张悦校　President：Zhang Yuexiao
副社长：蒋 涛　Vice President：Jiang Tao
编委会：黄长著　张悦校　陈洋彬　蒋涛　曾登高　刘江
Editorial Member：Huang Changzhu　Zhang Yuexiao　Chen Yangbin
Jiang Tao　Zeng Denggao　Liu Jiang

执行主编：孟迎霞　Executive Editor-in-chief：Meng Yingxia
编辑部主任：董世晓　Director：Dong Shixiao
责任编辑：高松　陈博　陈秋歌
Editors：Gao Song　Chen Bo　Chen Qiuge
特邀编辑：方梁　高昂　赵健平　吕娜　卢鹄翔
Contributing Editors：Fang Liang　Gao Aang　Zhao Jianping
Lv Na　Lu Dongxiang
美术设计：纪明超　Art Designer：Ji Mingchao
美术编辑：林象海　Art Editor：Lin Xianghai
流程编辑：白羽中　Coordinator：Bai Yuzhong
Tel：010-64351458
E-mail：editor@csdn.net

广告总代理：北京创新乐知信息技术有限公司
Sole Advertising Agency：Bejing CSDN Co.,Ltd
Tel：010-64376055
E-mail：ad@csdn.net
Marketing Dept：010-51661202（ext 149）
E-mail：market@csdn.net

发行部
Distribution Dept.010-64351431
E-mail：sales@csdn.net

读者服务部
Readers service Dept.
网上订购：http://dingyue.programmer.com.cn/
读者信箱：reader@csdn.net
地址：北京市朝阳区广顺北大街33号院1号楼福码大厦B座12层
Address: B-12th Floor Fairmont Tower NO.33 Guangshun North street,Chaoyang District,Beijing
邮政编码：100102
电话：010-64351436
传真：010-64348545

法律顾问：北京中润律师事务所 王杰
Law Consultant：Beijng Hengsheng Lawyer Firm
印刷：北京盛通印刷股份有限公司
Print: Beijing Shengtong Printing Co., Ltd.
出版日期：每月1日
Publication Date：the first day per month
零售价：RMB 15.00元　新台币 390元　HK $ 35.00（港、澳）
US $ 9.00（海外）
Retail Price：RMB 15, NT$390, HK $ 35.00,US $ 9.00

本刊文章版权所有 未经许可不得转载
发现装订错误或缺页，请将杂志寄回本刊读者服务部调换

106 基于Minitransaction构建可扩展分布式系统
高可扩展性的分布式系统几乎都采用了消息传递机制，消息传递虽然避免了共享内存容易产生竞态条件等缺陷，但在设计与实现时，需要处理大量复杂且易错的底层分布式协议。而Sinfonia最大优势在于将这些复杂协议的设计放到了更高的抽象级别中（Minitransaction），开发时只需关心设计相应数据 结构。

111 用云架构解决研发技术瓶颈

技术

113 漫谈Android系统的调试模型（下）
前两期文章介绍了Android系统支持调试的设备端设施和主机端设施，本期将介绍这些设施如何相互协作以提供各种调试功能。

117 移动设备的多核时代
随着移动设备上多核处理器的普及，越来越多的移动应用将通过并行化的方式提供更绚丽的3D渲染、更流畅的用户体验以及更丰富的特效，这对移动开发者的多线程编程技能提出了新的挑战。移动设备上并行编程时代的大门已经开启。

120 基于动态分类策略的ACL控制原理
ACL的重要体现在它决定了系统的灵活度，几乎可称为系统的基石，如何处理好这个问题非常关键。本文作者阐述了基于动态分类策略的ACL控制原理，不失为一种有效的解决方案。

程序人生

130 商务才是关键点
——专访6688.com创始人王峻涛

程序春秋

132 上善若水
——Cordell Ratzlaff引发的Aqua革命（下）

名人堂

134 软件传奇Dennis M. Ritchie
126 新书上架
128 Geek
136 程序幽默

企业专栏

26 IBM王小虎：架构师不是一日炼成的
62 Justep X5产品揭秘
78 三星bada2.0平台特性

世间再无乔布斯



我向来以果黑自居。原因很简单，我一般对人人都在谈论的东西心存疑虑，以前还看电视时，再大热的电视剧我也会选择在已经没人提起时再瞅瞅是否值得花时间。此外，我对会使人迷醉的东西也始终保持戒心。而这两个条件乔布斯和Apple的产品显然都符合。

但是，无论你喜欢与否，乔布斯的辞世给所有媒体人都布置了一篇命题作文。而本期，我们杂志的主题又是产品经理。我没有回避的理由。

评价乔布斯并不容易。他少年得志，早期经历有很多阴暗面，对自己私生女儿Lisa的所作所为最令人齿寒。年轻时个性上的缺点更多：脾气暴躁，动辄骂人开人，我行我素，喜欢出风头、将别人的功劳纳入自己囊中，对他人和事物非天才即脑残的二元论以及对待后者的残酷无情，对产业和产品的控制欲等等。总之，他曾是一个性格怪癖、很不招人喜欢的家伙。

1985年被逐出Apple的打击，1991年与贤惠而且睿智的Laurene组建家庭，2003年被诊断出癌症开始直面死亡，生活和时间似乎使乔布斯对人生、家庭的看法有了很大变化。他2005年在斯坦福大学发表的演讲感人至深，心态平和。为了孩子更多地了解自己，他主动请求Walter Isaacson为自己作传，曝光自己低调的私生活。他经常给许多媒体记者直接打电话，耐心解释自己的想法。他甚至在竞争对手Larry Page担任Google CEO后还抱病提供自己的建议。

他最后一次在WWDC做Keynote时，标志性的New Balance 993鞋不仅旧而且沾满了草渍，这完全不符合他极其注重细节的作风。John Gruber分析，草渍很可能是和家人一起散步（他患病后经常与人一同散步）时留下的，在生命的最后时期，他已经对这些细枝末节视而不见。

有一点是毋庸置疑的，乔布斯是我们时代最优秀的产品经理。他是依靠产品颠覆多个行业，并进而改变世界的。戈尔说，乔布斯的最大产品其实是Apple公司本身，而这家公司已经雄踞所有竞争对手之上。

他具备通常产品经理应有的各种素质：对业务和用户需求的深入而且超前的理解，足够的技术素养，用户体验和设计方面天才般的直觉和判断力。关于他的技术能力之前一直有争议，Eric Schmidt的回忆给予了证实，他能够与软件工程领域博士出身、lex作者之一的老施争论面向对象和Objective-C的细节并压倒对方，这非常人能够做到。当然，这些看上去不同的素质，其实归根到底，都是一种化繁为简、抓住本质的思考能力。

他真正卓然超出一般IT产品经理的地方在于，罕见的激情和远见卓识，将文化和情感带入产品之中，达到了全新的境界——开发用户渴望拥有的产品。他对完美的追求，他的理想主义，在这样世俗的商业社会中能够取得成功，也激励了更多后继者。

终其一生，他争议最大的地方，可能是坚定地走一体化、垂直整合路线，从而不可避免地站在了封闭的一边。一方面，App Store模式为世界各地的开发者创造出史无前例的有利可图的全球化大市场，众多高质量的软件为数以亿计的普通消费者带去快乐和便利。另一方面，封闭和控制，在很多人眼中也意味着通往奴役之路。自由软件之父Richard Stallman完全负面的评价虽然显得无情而且不合时宜，却并不难理解。对此，开源软件的提出者Eric Raymond有更深入的阐述，他说，乔布斯最可怕的地方在于，他的封闭花园看上去那么酷，使得从者如云，不仅心甘情愿地接受控制，而且由衷地赞美，这种文化力量比当年盖茨代表的商业帝国更甚。

*Wired*杂志的创刊主编Kevin Kelly在2007年就提出了the One Machine的概念，认为最终会出现统一的信息系统，处理能力超过全人类，人反过来成为它的扩展感官。如果这种图景成为现实，而且是以封闭花园的模式实现，那么人类的未来的确堪忧。

我不知道乔布斯对此怎么看。斯人已去，知之罪之，对他都不再重要。他已经做了一生钟爱的事情，而且做得很不错，去时有深爱他的家人陪伴，这已经足够。

邮件：liujiang@csdn.net
新浪微博：@刘江CE
欢迎大家交流反馈，欢迎投稿、批评、建议和挑错

游戏与社交

互联网游戏潜力巨大，但无论社交游戏还是移动游戏，无一不是对手环伺的竞赛场。Zynga依附于社交网站，赢得了成功，也获取了众多效仿者。在中国，谁是Facebook最好模仿者的讨论未休，自家平台上搭载社交游戏已然成为标配。

对于游戏公司来说，创新是维持高营收的唯一手段，它们无法像石油、硬件公司一样吃老本，一旦用户兴趣寡然，盈利能力下降不可避免。据统计今年第二季度，Zynga净利同比跌幅达95%。其他众多小游戏平台，如不能持续创新，一味扎堆追逐热点，未必能过得更好。

Zynga自身的努力是生产新的产品，并推出“Zynga Driect”新服务，减少对Facebook的过度依赖。此番另立门户，必将引发社交平台震动，对于Facebook而言，虽然拥有绝对庞大的用户群，但后来者追赶的势头也不可小觑。关停失败的Buzz，所有产品全力整合进Google+，Google进入社交领域的脚步就从未停止，其圈子的概念获得专家好评，有类似的社交网站未及上线即获得VC千万投资，从侧面反映了业内对社交网站的信心。

然而，开放注册使得Google+的用户激增后，用户发帖量短期迅速下降41%，IT专家对此褒贬不一。让用户找到支持它的理由，对Google+才是重中之重。

未来互联网的入口



10月4日，苹果在其加州总部发布了新一代手机：iPhone 4S。与iPad发布时的情形相仿，人们最先关注机器的外形，失望的声音不绝于耳。但人们很快发现自己开始对一个新功能有了特别的兴趣：Siri，可以让用户通过语音利用iPhone多种功能的软件。

Siri源于由美国DARPA资助的人工智能研究项目之一CALO，主要由斯坦福研究院承担，项目目的是开发有一定认知力、具有自学习和自组织能力的个人助手系统。原Siri公司的联合创始人兼工程副总裁Adam Cheyer是CALO项目的首席架构师，CTO Tom Gruber是斯坦福大学的人工智能、语义网专家。除了核心的智能算法之外，Siri还是一个自然用户界面系统，而且服务器端可以整合来自大量开放API的数据。

有专家撰文称“如果你错过了Siri，你就错过了整个网络”，并将它与1990年代“通过域名找页面”、2000年代“通过页面交互聚人”相提并论，称之为第三代互联网交互模式：“消除网页，将API与人直接联系在一起”。

对于那些在iPhone 4S上使用过Siri的人来说，它是一个改变游戏规则的技术：它知道如何将人类语言转化成有针对性的API调用，从潜在的不断扩大的数据库中拉取正确的信息。

据称Siri不仅可以装在iPhone手机上，还将安装在苹果的笔记本和TV上，如果Siri得到了广泛应用，基于人类经验而建立的数据库最终将能够解决很多用户每天面临的问题，而人们查询互联网首先依赖的将是Siri。

华尔街的中国互联网

对VIE的一番言论，让马云获得几乎一边倒的负面评价。所谓新政的影响是投资者认为VIE存在风险，连锁反应是“中概股”全线下跌、新浪暴跌15%、搜狐跌5.6%、百度跌3.9%、网易跌3.9%、盛大跌1.2%。

因坊间传言，运营微博需要牌照，并实行实名制，而颇具人气的新浪微博却不在牌照发放名单里，新浪美股市值一夜蒸发10.25亿美元。10月中旬，新浪官方召开积极运用微博客服务社会经验交流会。有关部门领导讲话鼓励政府机构和人员开通微博，主流媒体对此也予以报道。随后新浪股价暴涨18.35%。

在美上市的中国互联网表现低迷已持续近半年。10月，美国司法部门介入了对部分在美上市中国公司的财务违规调查，受此影响，百度跌了9%，新浪跌了10%，优酷跌了18%。

中国互联网的股票就像坐上了过山车，从高峰到低谷，再从低谷到高峰，究竟方向如何只能由铁轨决定。

告别了年初互联网公司纷纷登陆美股的火热，如今在美IPO需要更多的考量，自7月起，一直犹豫不决的迅雷，终于在10月中旬决定推迟上市，分析称是出于盈利、版权等多方考虑，但塞翁失马，焉知非福。

“我们并没有看到针对Android的专利攻击手段已取得效果。我们认为，其他公司所采取的此类行动，最终将疏远客户……我们此前宣布了收购摩托罗拉移动的动机，我们将重视对Android生态系统的保护。”

—— 一位分析师向佩奇提问，Google如何保护Android生态系统的良好发展？佩奇如此回答。

“Google+是可悲的马后炮，在推出之时它完全没有API，上回我检查时，只有可怜的一个API请求。”

——Google软件工程师Steve Yegge公开发表文章，后来文章被撤销。

“开设一家像Facebook这样的公司，或是开发一款像Facebook这样的产品，需要决心和信念。所有值得做的事都是十分困难的。”

——扎克伯格认为成功没有捷径。

“在我们浏览网页时，会有很多看不见的网络狗仔、cookie和其他数据搜集器如影随形，将我们的一举一动报告给营销企业，然后搜集并组建我们的在线活动资料库。”

——美国联邦贸易委员会主席Jon Leibowitz表示互联网的隐私泄露现状远比网民想象得更为严重。

“正当中国开始与世界其他地方一样讨论乔布斯的遗产之时，很多中国人却发出疑问：中国为什么出不了熊彼特式的企业家，出不了自己的乔布斯、比尔·盖茨和马克·扎克伯格？”

——美国《福布斯》杂志网络版发表署名Panos Mourdoukoutas的文章称，中国之所以出不了自己的乔布斯，一方面是因为传统的儒家文化使然，另一方面则源于当前经济体系的缺陷。

“小米的时间肯定是最多的，毕竟我是创始人，也是CEO，金山那边我是董事长，对于董事长来说，最重要的是找一个合适的CEO。至于投资，那是个业余爱好，我正考虑组一个专门的团队来打理。”

——雷军在回答媒体关于在小米、金山和投资等之间，个人的时间和精力如何分配的问题时，如此回答。

“我建议你去四川，告诉那里的人们不要去IT商场里购买PC，因为后PC时代已经到来，然后听听他们对此的评论。”

——宏碁全球副总裁兼中国区总裁艾仁思回应媒体关于“后PC时代”的疑问。

CSDN十大资讯

2011年9月26日~10月26日

01 苹果宣布前CEO乔布斯辞世

10月6日，乔布斯辞世，终年56岁。苹果董事会在声明中称：“史蒂夫的才华、激情和精力是无尽创新的来源，丰富和改善了我们的生活。”乔布斯用一系列创新产品证明了优良的设计比高科技本身更有吸引力，改变了人与科技互动的方式。

02 C语言之父Dennis Ritchie辞世

10月9日，Dennis Ritchie去世，享年70岁。他发明了包括C语言在内的多种编程语言，并研发了Multics和Unix等操作系统。无论哪一项，在计算机发展史上都有着举足轻重的地位。自20世纪70年代起，Dennis Ritchie的工作就收获了包括图灵奖在内的诸多奖项与表彰。

03 微软回应用户不满，称将改进Win8 Metro界面

10月13日，面对很多不满意Windows 8 Metro界面的用户，微软作出回应，表示将优化部分界面以平息抱怨。Windows Live部门总裁Steven Sinofsky说：“这些用户正在犹豫是否选择Metro界面。这是一种自然的反应。”但在与用户的争辩中，微软也明白需要作出改进。

04 10个管理工作时间的小技巧

作者 / Joe Mathews，Don Debolt，Deb

这篇文章谈及普遍的业务流程是由想法、交流、行动三个方面组成，在日常工作要提前做出计划，才能确保工作过程中的顺畅，成为自己时间的掌控者，于是作者用六条原则从不同角度强调计划先行，其余四条原则则重点阐述一次只做好一件事的重要性。

05 后乔布斯时代：苹果仍有可怕潜力

作者 / 刘江

乔布斯去世后，苹果走向令人瞩目。在Tim Cook的带领下，苹果会继续推出创新性的产品吗？作者综合多方资料指出，苹果将会推出性能更出色的芯片，并使用闪存取代硬盘，加之如Siri的人机交互创新，具备真正颠覆Web的力量。

06 Windows 8将带来什么

作者 / 葛鑫

借助Windows 8，微软意图孤注一掷地将PC与移动终端设备统一，虽然微软的战略处在矛盾之中，用户选择也难以预测，但无论成败与否，这一决定都将左右微软和IT界的未来。事实上，很少率先开发新技术的微软，总能够上演后来者居上的好戏。



07 提升员工士气的七大法则

作者 / Marcus Erb

作者分析筛选国外公司的成功实例，举出7个提升士气的方法。其中包括，让员工感受他们的工作不仅仅是一份工作；举办庆功宴；让员工发展个人项目；遵循公司惯例；不要忘记娱乐；培养员工积极向上的态度；偶尔离开办公室，服务社区。

08 到底什么是产品经理？

作者 / Martin

产品经理是怎样的职位？主要职责是什么？大家多少对此有所困惑。作者的解答是，产品经理是商务、技术和用户体验三个岗位的交集。他的职责是定义产品的本质属性、为客户的问题设计解决方案、和所有成员一起完成工作。整个过程，艰辛而有趣。

09 C/C++程序员的编程必备“良方”

作者 / 詹杭

C++面向对象中的繁琐机制，让程序员们在编程中频频“犯病”，本篇博客文章中，作者介绍了C/C++编程中，需要注意的20个问题，从类的设计到函数的变量声明，从继承类型到继承层次，巨细靡遗，都是经验所成。

10 Facebook上市将终结互联网高估值时代

作者 / Dave Whorton

作者是Tugboat Ventures创始人，他在文章中称，Facebook很可能在2012年IPO，这意味着互联网公司高估值的终结，该行业的滑坡很可能在其上市后发生。文中建议创业者脚踏实地地努力，专注于取悦用户，建设高效的核心团队。这才是硅谷永久的成功公式。

全球50年最伟大科技发明

科技网站T3.com投票评出了过去50年来最伟大的科技发明，苹果的iPhone位居第一，因其风头正劲，拔得头筹不足为奇；但20世纪80年代出现的索尼随身听力压iPod位列亚军，不会褪色的科技魅力令人叹服。负责此次调查的T3.com编辑Kieran Alger表示：“30年后，索尼经典版随身听依旧被视作超过iPod的最伟大创新，这也证明索尼磁带式音乐播放器极具开创性。”本次榜单中，苹果产品独占三席，网站占得两个名次。

01 iPhone，2007年，Apple

创新地引入了多点触摸；将重力感应、陀螺仪等引入手机系统；与应用商店紧密结合，颠覆了智能手机行业。

02 Walkman，1979年，Sony

本是Sony的一个品牌，后成为随身听的代名词，刚发布就风靡全球，创造了耳机文化，催生无数新型音乐设备。

03 iPod，2001年，Apple

优雅的外形秉承一贯的苹果风格，拥有独特和人性化的操作方式以及巨大的容量，网上商店革新了音乐行业。

04 Windows，1985年，Microsoft

世界上用户最多、兼容性最强的操作系统，存世26年来，不断演进，使得个人电脑的概念深入人心。

05 iPad，2010年，Apple

史上最成功的平板电脑，集便携性和功能性于一体，全面融合了苹果公司的既有平台优势，开辟了全新的电子产品领域。

06 YouTube，2005年，Google

全球最大的视频分享网站。受益于技术进步、宽带发展和便携摄影器材的普及，使得用户贡献内容的视频大行其道。

07 Trinitron TV，1986年，Sony

又称单枪三束管，画面比起同时代的普通显示器颜色更加鲜艳锐利，成为CRT显示器最佳品质的同义词。

08 Sky+，2001，British Sky Broadcasting

数字式录像机，通过内置此设备的机顶盒录制、暂停或回放电视节目，开辟了互联网与数字技术互动功能的先河。

09 Facebook，2004年，Facebook

拥有5亿多用户，是世界最大的社交网站，作为Web 2.0精神的代表，被看作是下一个Google，未来互联网的入口。

10 VHS，1976年，JVC

作为家用录像机的标准格式，成为盒式录像带的代名词，令录影、录像普及到平常百姓人家，同时发展了影像租赁行业。

摩托罗拉发布RhoElements HTML5框架

此为Web应用框架，基于摩托罗拉7月刚刚收购的Rhomobile平台。摩托罗拉称有计划在未来几个月为合作伙伴和客户发布跨平台的开发工具、企业集成组件以及基于云的服务。 RhoElements提供了HTML5/JavaScript/CSS应用框架，支持HTML5特性（包括，缓存、Web存储等）。企业客户可利用RhoElements开发和发布基于摩托罗拉嵌入式设备的应用。



Ubuntu 11.10 正式版发布

Ubuntu 11.10（代号Oneiric Ocelot）正式发布。新版本除了带来新的内核、软件、体验外，还带来了更成熟稳定的Unity和Unity-2D。新特性包括：Lenses和界面的改进；软件中心5.0；新的ARM架构的镜像；新的DVD形式；新的应用开发者网站：developer.ubuntu.com；新的本地化ISO工具，以及若干应用程序更新。



JBoss Seam 2.3 Alpha发布

虽然Seam 3已经出现在人们的视野里，但应用范围和成熟度远不如Seam 2。最近发布的新测试版本Seam 2.3 Alpha，主要变化包括：项目基础构建迁移到Maven；支持 JSF 2；支持JBoss 7和最新的Arquillian测试框架（一些例子包括JBoss 7配置）。



QtWebKit 2.2正式版和Qt Creator 2.3.1发布

QtWebKit基于开源的WebKit引擎，提供在Qt中使用Web浏览器引擎的功能，包含C++ API，允许在Qt应用程序中嵌入网页内容，并可以检验和操作。QtWebKit具有渲染HTML、XHTML和SVG文档，使用CSS排版，运行JavaScript等功能。相对于之前的版本，改善了音频和视频支持，添加了对新标准的支持，修复了WebKit代码库中的安全问题和其他Bug。

此外，Qt官方还发布了Qt Creator 2.3.1版本，该版本修复了一些关于项目管理、QML/JS调试、Qt快速设计器、Symbian、远程Linux支持方面的问题。



Google推云存储服务和云端数据库

Google推出了Cloud Storage云存储服务，能让用户以低成本将业务数据存储到网上，并在任何地方都能够访问数据，节省了高昂的服务器管理费用。目前，这项服务面向中小企业以及大型企业开放。它的前身是面向开发者提供的存储服务Google Storage for Developers。此外，Google还发布了云端数据库Google Cloud SQL，是一个接近传统数据库的云端服务，目前只支持用在Google App Engine上。该服务也提供数据导入和导出功能，所以可将现有MySQL数据库迁移到App Engine云端，目前免费。

Apache Subversion 1.7 正式版发布

这是Subversion并入Apache基金会后的首个主要版本。该版本包括了大量新特性、Bug修复及其他改进。Subversion是开放源码、多用户的版本控制系统，支持非ASCII文本和二进制数据，支持可在本地访问或通过网络访问的数据库和文件系统存储库。相对于传统的RCS、CVS，Subversion采用了分支管理系统，其设计目标就是取代CVS。互联网上免费的版本控制服务多基于Subversion。在2009年11月，Sbuversion加入Apache软件基金会。

Google推出JavaScript测试工具

Google推出JavaScript测试工具Google JS Test。Google JS Test是一个运行于V8 JavaScript引擎下的JavaScript单元测试框架，在Google内部负责测试Chrome的快速JavaScript执行速度，现在Google以开源工程开放给大家使用。



AppMobi推出新XDK

AppMobi发布了新工具PhoneGap Mobile App XDK，可供开发人员创建、模拟和测试PhoneGap项目。该工具作为IDE，提供了用于创建HTML5和PhoneGap应用的全套开发功能。新XDK为PhoneGap项目开发提供了直观的开发环境。



KDE发布Plasma Active

这是专为平板设备设计的用户界面，与Android、iOS或webOS不同，Plasma Active不是操作系统。它是KDE 4.x类型的界面和API接口。根据Plasma Active官网介绍，Plasma Active是创新的针对智能用户体验的技术。它适用于所有类型的平板、智能手机及其他触摸计算设备如机顶盒、智能电视等。此开源项目目的在于：用最小的内存需求提供更快的用户体验；定制和模块化，支持不同架构；适应用户操作行为的界面。



Communications of ACM

2011.10

数字化生存之道

如今的技术为人与人之间的沟通交流创造了新的机会，然而重度的技术依赖会对人类的行为产生何种影响，研究者对此的兴趣与日俱增。

很明显，与人交流是人类的本能。研究表明，一个状态良好的人会在社交方面花费更多时间。然而身处一个数字化的时代，沟通的含义，甚至友谊的定义都在发生急剧的改变。

人类之间的联系越来越多地依靠数字设备。大约有20亿人使用互联网，5亿人使用移动电话。并且有越来越多的设备可以提供先进的计算和通信能力。无论是在珠穆朗玛峰顶还是在遥远的南太平洋环礁，人们不会失去与外界的联系。

在美国，13～17岁的青少年平均每天会发送110条短信。人们无法忍受接触不到任何数字产品，其使用频率也与日俱增，花费的时间也在增加。研究者正在探究这种增长所产生的影响。“我们正在经历越来越多地沉浸到数字媒体的过程，这成了生活的首要问题”，斯坦福大学的传播学教师说，他的著作为《欺骗电脑：在人际关系中，机器教会了我们什么》。

“数字沉浸”的影响是什么？将会多大程度上改变人们的思维方式和行为？从而改变我们看待世界和与人交流的问题呢？关于这个问题的复杂性，研究人员也是刚刚意识到。通过技术，人们正以前所未有的方式彼此连接，同时也产生了新的问题和麻烦。

毫无疑问，数字技术正在侵入人们的生活。爱立信在2011年的一项调查中发现，35%的iPhone和Android用户早晨时会在床上查看邮件或是Facebook。40%的受访者，睡觉前都会使用手机。普通美国人每天有2.5～3.5个小时使用数字设备。尼尔森的报告称，社交网络、在线游戏和电子邮件是人们使用数字产品的主要原因。

“数字沉浸”如此诱人，原因也许是人类用它弥补自身的弱点，但同时也证明了自身的脆弱。人类个体总是孤独的，但同时又对亲密关系充满恐惧，不间断彼此沟通会提供有人陪伴的假象，但又没有友谊的义务。对于远距离和大量的人际交往，绝大多数人是无能为力的。然而数字技术的使用使得人们不必为面对面的沟通掌握太多关于耐心、节奏、敏感的交际技巧。我们变得更加适应数字媒体，随着时间的推移，我们越来越难以接受复杂的东西。结果是人们抱怨自己太忙于与人沟通而没时间思考、创造。最后的矛盾是我们热衷于与人沟通，而忽略了值得我们关心的人，虽然人与人不停地接触，但我们只是孤独地在一起。

很多人承认，相比直接与人沟通，留一通语音邮件让他们觉得更轻松，发短信节省了时间和电话中可能的承诺。我们应用技术与人取得联系，想象着对自己的时间和隐私有更多的掌控，但此种安慰的结果是我们疲于应付各种碎片的关系，而变得过于劳累。

“数字沉浸”的影响还表现在其他方面，比如学习。学生在把注意力放在课程和学习上有了更多的问题。学生的顺序、逻辑和理性的思考能力似乎受到严重的损害。事实上，人类无法处理多任务，而只能在任务之间来回切换，数字生活导致碎片化的思考使得人们不得不承担繁重的“多任务”，减少了短期记忆，换言之，我们是健忘的。我们越来越难以控制再次检查收件箱的冲动，难以控制点击在线广告的冲动，虽然闪亮的电子产品能够满足物欲，却无法慰藉灵魂。在时间用金钱来衡量的社会，电子产品必不可少，但也许不可避免的是，它们也正成为我们的控制者。

研究表明，重度技术依赖会带来高水平的孤独感和抑郁症，越来越多的科学家认为是该重新控制技术和我们的生活了，越来越受欢迎的概念是，关掉电子产品。2010年，安息日宣言项目出现，它提倡“每周一天不连线”，提醒我们必须主导技术，而不是为其所控。

改进脑机接口

研究人员正致力于改善脑机接口（BCI），以帮助身体瘫痪的人更有效地与人沟通，适

应环境。

科幻小说早就有人类和机器在思维方面沟通的假想，不仅是让电脑模仿人脑的机制，而且可以弥补大脑本身在信息处理和访问数据方面的短板。然而与虚构故事中增强健康人的人机接口不同，现实中的科学家们已经在脑机接口方面取得了创新，可以弥补一些残障人士缺失的基本功能，如视觉、听觉和运动。

BCI研究从神经科学、计算机科学、物理和电气工程等领域吸取了大量成功，使得受限于传统方法的医疗解决方案有了新的进展，其中包括对帕金森症患者的脑深部电脉冲，失聪者的耳蜗移植。通过BCI（侵入式的和非侵入式的），瘫痪的人与外界交流有了更有效的方式。

然而科学家们在此领域遇到的挑战也是巨大的，包括硬件传感器、数据处理系统等方面遇到的难题都需要来自不同领域的专家共同解决。“解决这些问题并非不可能，我们建立起了一个协作紧密的实验室”，克里夫兰研究所的Taylor说。目前她的工作重点是使用大脑信号引发脊柱损伤者的运动功能。

“我们通过记录患者的大脑活动，将信息解码传递给手臂”，Taylor说，“一方面有效地解码大脑的运动意图，另一方面成功地刺激瘫痪的肌肉产生运动，就像患者自己这么做一样。”

依靠植入脑部的芯片，科学家可以记录脑部信息处理的过程，并且加速智能算法，以适应大脑的运算能力。与大多数神经网络的应用不同，Taylor所发明的新算法需要在接受输入的同时解码，以及时地产生肌肉活动。她称这种算法为“适应性算法”，并强调在任何平行适应性系统中，找到一个适当的适应性是确保稳定改进的关键。BCI相关技术的成熟还有很长的路要走。

20世纪90年代末，大多数BCI研究组织都在编写定制的软件，但因为没有共同的平台，限制了研究成功的交流，不过BCI2000的出现改善了这一情况。它的核心是一个通用的软件系统，支持多种数据采集方式和反馈模式，并且允许与外界程序进行交流。目前这款系统是免费、开源的。

BCI潜力将使得人机交互的方式发生转变，但大多数BCI研究者提醒说，该领域仍然相对年轻。“在一个真正理想的未来，应用BCI，人类将告别身体残疾，残疾人与外界的互动将和正常人没有分别。”

计算新闻学

计算新闻学可以定义为在新闻领域计算的应用，诸如新闻资讯的采集、组织、沟通和传播，同时要保证新闻的准确性。

现代计算机收集和传播信息的能力看似是无限的，新闻工作者将如何利用这些能力加强自己新闻的分量呢？新闻搜索、社区讨论、微博以及互联网每天产生的万亿字节的文本信息都是巨大的潜在新闻来源。市场营销专家、社会学家、信息专业人员和政府都在投入巨资，应用创新算法分析这些数据源。

新闻记者也不应该错过这样的新手段，包括如何探索新方法、信息来源和如何与社区沟通以及如何管理自己的信息。计算的新闻可以改变发现故事的角度、呈现方式、聚合手段、商品化途径以及存档的方法。计算的运用可以以创新绘图、主题检测、视频分析、个性化聚合和可视化等手段更好地呈现新闻。

新闻记者和计算机科学家一方面正在开发新的方法来介绍深度报道的成本和困难，另一方面也面临着如何把纸质出版物转变成可检索的数据库以及如何利用现在的计算接口更好地适应新闻报道的特殊需求。并且他们还面临着文化上的挑战，作为计算机科学家处理信息的方式如何与新闻记者生产新闻的方式契合，对两者来说都是需要克服的障碍。

其实新闻记者与计算机技术打交道的历史由来已久。20世纪的美国就有记者应用取样和相关分析的方法，得出学校考试成绩和学生收入的关系，从而揭露出学校和老师在此方面的作弊行为。20世纪80年代后期，关系型数据库和SQL的出现，记者们则可以在交叉引用系统中找到新闻点。

如今新技术的产生，更使得新闻工作者能够获得大量非结构化的文本（金融股票的表格、法庭纪律、政府日历），并通过透视文档内部的关系，以较低的成本发现和发布新的调查报道。

当然这只是一个开始，未来记者将和他们的伙伴、计算机科学家、应用开发者和硬件工程师一起，寻求符合公共利益的信息，改善社会。

Wired

2011.10

窥视Google+

对于Google的成功，人们已经有了深刻的印象。但当涉及到社交网络，一次又一次的尝试却又彻头彻尾的失败。名为Orkut的服务从一早期就显得默默无闻（除非你住在巴西）。复杂的Wave让大多数用户感到困惑。Buzz试图通过用户的Gmail联系人建立熟人网络，也未收到成效。

Buzz是个尴尬的崩溃。没有取得用户同意的权限操作，造成客户和评论家一致的关于隐私的抨击。尽管Google解决了这些问题，但最初的污点难以消除。更糟糕的是，其最初的缺陷导致了联邦贸易委员会的审查。最终，Google同意提交未来20年的隐私监督。

如今Google正全身投入Google＋中，一个全新的更宏大的社交服务。将包含信息



的共享、交流和Google的产品（Gmail、YouTube、地图等）全面融合。不像Facebook和Twitter，每次更新通常都会与好友共享，Google+设计成方便对特定群体分享，除了类似Facebook的朋友、家人和同事的更新外，Google+还包括一个“Interesting Stream”推荐用户可能喜欢的主题，并且能够立即创建一个最多十人的视频会。

意识到过去的失败，Google也对Google+持保留态度。但用户的热情很快打消了这一点。才发布没有几天，就有10万人注册了该服务。虽然这项服务尚处于发展阶段，但Google产品副总裁Bradley Horowitz表示，它最终可能会改变整个公司。

在接受Wired杂志的采访时，Horowitz谈到Google+刚一推出时，整个团队花了大量时间来处理意外的情况。虽然最后得到了正面的反馈，但他们会更留意于长远的发展。

目前为止，Google发展了很多项目，并且每一个项目独自发展，以至于有时候会偏离Goolge的方向。但Google+就是Goolge自己，搜索、广告、浏览器、手机系统等这些都将服务于它。

在外部看上去，Google+的活跃度不高，但Horowitz表示正是吸取了以往的教训，Goolge+极度重视用户的隐私，每个人可以建立自己的小圈子，这些内容是公众无法看到的。另一方面，也有一些过分活跃的用户产生烦恼他人的噪音，Horowitz说他们已经在制订计划，解决信息过冷或过热这两端的问题。

对于Google+的未来，Horowitz坚信不会重蹈Buzz的覆辙。

记忆工程

还记得一年前的今天，自己做过什么吗？住在印度尼西亚雅加达的社会媒体专家Daniel Giovanni就可以。最近他开始使用一个叫4SquareAnd7YearsAgo的不错的服务。这是一个服务插件，可以整合进Foursquare插件里，那些地理位置的标签能够显示用户吃过的餐馆、喝过的酒吧和社交场所。每天早晨，4SquareAnd7YearsAgo都会在一年前的今天导入用户的签到，并且发送电子邮件给用户一个总结。

这个插件每天都会给用户一个奇特的对过去的缅怀，让人感到有趣。我们可以跟随Giovanni回顾一年前的7月20日发生了什么。根据电子邮件里的回顾，那天他的毕业论文答辩。他在早晨7时42分抵达学校做准备，中午12时42分离开，然后去看电影和朋友们一起庆祝毕业。过去的事情就在这一瞬间回滚到他眼前。

“这个应用可以帮助你重塑对生活的回忆。”Giovanni说，“4SquareAnd7YearsAgo是一个趋势的例证，这被我称为‘记忆工程’，我们将过去的事情数字化，从而塑造成我们有用的记忆。”

眼下，我们的数字生活是那么臃肿庞大，产生的信息是难以估计的。我们中的许多人每天产生大量的个人资料，包括手机拍摄的照片、文字信息、状态更新等。通常情况下，我们每个人都在成为生活的记录员。但几乎没有人会回过头去看看这些东西，因为他们生硬、枯燥，看起来让人昏昏欲睡。

“记忆工程”的工程师们正在解决这一问题，通过创建格式化的服务，以一种机智而且往往是艺术的方式处理我们的生活数据。4SquareAnd7YearsAgo是纽约的程序员Jonathan Wegener在2010年冬天发布的。他们先是与Foursqure服务结合，然后又着手推出了PastPosts，这也是同样原理的软件，只不过这次是结合Facebook上的服务。短短几个月内，它就已经积累了数以万计的用户。

“我们有很多方式在世界上留下自己的印记”，Wegener说，“我想做的就是使它们再次对人有意义。”

Wegener提醒说，如果你痴迷于存储和组织个人的生活记录，不要忘记一件事：不要忽略细节，不然“过去”只是一堆在地图上的点。目前，很多“记忆工程”的情节集中在个人经验上，比如采取照片、声音等，使之成为一个可追述的日记。



我们时代的凯撒
史蒂夫·乔布斯

“我来到，我看到，我征服”

——恺撒

史蒂夫·乔布斯（Steven Jobs），生于1955年2月24日，故于2011年10月5日，享年56岁。

苹果公司将官方网站首页换成乔布斯大幅照片，以及“1955—2011”字样，并发布消息说：“苹果失去了一位富有远见和创造力的天才，世界失去了一个不可思议之人。”

当后来者回顾乔布斯的一生，会发现他是一个自身充满矛盾的人，睿智却粗鲁、温和却独裁、品位独特却能引领潮流。他缔造了多款流行电子产品，颠覆了计算机、娱乐等诸多行业。如果现世的每一位伟大人物，都能比做一位先贤，或许我们可以将乔布斯称作“我们这个时代的恺撒”。

iCame

在20世纪50年代，做单身母亲还是一件丢脸的事情，无奈之下，乔布斯出生才一周就被生母送予他人抚养。养父母给予了乔布斯良好的教育，但幼年的乔布斯并未从同龄人那里收获到什么。他是个孤独爱哭的男孩，当时的同学这样回忆乔布斯。

然而乔布斯找了自己喜欢做的事，开始对计算机着迷。“在硅谷长大，我很幸运”，乔布斯如此回忆自己的青少年期。10岁多的时候，他在NASA Ames研究中心第一次见到计算机（不是个人电脑，而是大型的终端机）。

高中时，乔布斯结识了后来的创业伙伴沃兹尼亚克（绰号沃兹）。当时两个人开始制作一些新奇的产品。那时的乔布斯就知道了自己的身世，更增加了他要有一番作为的决心。沃兹有实际操作的技术，乔布斯具备极佳的演示产品的天赋。那时沃兹设计了一款拨打长途电话的装置，而乔布斯负责把它高价卖出去。

但后来的乔布斯很快对此失去了兴趣。1972年，他说服父母送他去里德学院念书。一

个学期后，又因为实在感到无趣索性退学了，但他仍然借住在朋友宿舍，在里德学院旁听字体美术设计等课程，也因为对书法美术的素养使得他对电脑软件字形美学处理特别重视，这也使得苹果电脑日后吸引了许多美术设计工作者来使用。

1974年打扮得像个嬉皮士的乔布斯来到雅达利做一名技术员。但很快他就只身前往印度灵修。在印度的经历无疑对乔布斯是一次洗礼。回来后，乔布斯开始以一种与以前不同的方式从最基础做起。

年仅 21岁的乔布斯找到了从前的伙伴——26岁的沃兹，两个人在乔布斯家的车库里成立了苹果电脑公司。乔布斯曾说过，苹果公司就是他全部的事业，至此一项伟大的事业开始启航。

iSaw

年纪轻轻的乔布斯和沃兹制造了世界上最早商业化的个人电脑之一，并称为Apple I。紧接着，在沃兹自家的厨房里，Apple Ⅱ诞生了，销售量突飞猛进。1983年，苹果公司的业务越做越大，乔布斯发觉自己在很多事情上的能力还不足够，便开始寻找出色的CEO管理公司。他找上了百事可乐公司的John Sculley，便用 “你是想卖一辈子糖水呢，还是改变整个世界？”的话语劝动了他加入苹果公司担任CEO。

1984年，首台Macintosh电脑推出，苹果公司打出电视广告《1984》，广告模仿乔治·奥威尔的著作《一九八四》，引起很大回响。已到而立之年的乔布斯此时可谓春风得意。然而意想不到的打击随之而来。

1985年，电脑业界普遍萧条，Sculley认为乔布斯已成为公司问题的来源，在其自传《苹果战争》中他认为乔布斯并不具有技术研发的能力，并把乔布斯赶出了苹果公司。在随后的许多年，乔布斯对此一直难以释怀。

iConquered

离开苹果公司后，乔布斯于同年以700万美元创立了NeXT电脑公司，并发展出NeXT电脑及NeXTSTEP操作系统。一年后，面对缺乏现金及周转不灵、公司并没有新的产品面世的境况，他呼吁创业投资者。最后，他得到亿万富豪佩罗·罗斯的关注，后者投入大量资金给他的公司。1990 年，NeXT首先发售价格9,999美元的工作站电脑。因为价格过高，未能成为流行的电脑商品。但这次失败的教训却让乔布斯有了日后返回苹果的资本。

1986年，乔布斯花1000万美元从乔治·卢卡斯手中收购了Lucasfilm旗下的电脑动画效果工作室，并成立了皮克斯动画工作室。之后十年，该公司成为了众所周知的三维（3D）电脑动画公司，并在1995年推出全球首部3D立体动画电影《玩具总动员》。该公司已在2006年被迪士尼收购，乔布斯也因此成为最大的自然人股东。一位科技领袖就这样以一种曲折的方式改变了动画行业，进而是娱乐行业。

1996年，陷入财政困难的苹果电脑以4亿美元收购了NeXT电脑公司，同时乔布斯也回到了苹果电脑。

1997年，当Gil Amelio离开公司后，乔布斯重掌苹果电脑的大权。在同年推出iMac，创新的外壳彩色透明设计在美国和日本大卖，使苹果电脑度过财政危机。并在之后推出深受大众欢迎的Mac OS X操作系统、iPod、Apple TV、iTunes Store、iPhone、iPad等一系列广受市场好评的产品。

2011年3月2日，乔布斯出席了iPad 2产品发表会。此时的他已被确诊胰脏肿瘤，并接受过一次肝脏移植手术。

iQuit

乔布斯在演讲台上挥洒自如，在生活中却相当低调，十分注意对自己家人的保护。他曾说过，是否成为亿万富翁也许真的不那么重要，对他而言，真正重要的是内心平静地和家人道晚安。在他生命的最后，也正是家人陪伴在他身边。

“伟大如我凯撒者，死后也是两手空空”

——恺撒

捆绑出来的泰坦

推出了Oracle Big Data Application这样全方位的产品，Oracle这次尝试是否成功呢？

Oracle在旧金山举行的Oracle OpenWorld大会上发布了一个令业界眼前一亮的产品——Oracle Big Data Application。Big Data不同于我们常说的关系数据，它可以是Blog、邮件、视频、地图，也可以是标识生物信息的指纹、声纹、血液数据，抑或是其他各种类型、各种新式的非结构化信息。

随着互联网内涵和外延的拓展，这些数据不仅被Facebook、Baidu、Google、Amazon这样的互联网公司大量使用，传统的企业应用中也越来越多地出现它们的身影，如果采用传统的关系数据库管理和使用它们就好比“笨贼偷石臼”，不仅徒劳无益而且也非常不经济。而Oracle之前给业界的印象一直以关系数据库领导者的形象居多，尽管这届Oracle OpenWorld大会前Oracle已经通过一系列官方声明和产品发布为这个产品诞生做了很多铺垫，但当Oracle真正把一批产品捆绑在一个“铁皮盒子”里一起销售时还是令业内陷入了一场争论——Oracle下一步到底要做什么？

我们先看下Oracle Big Data Application包括：

- 用于开发和运行海量数据应用的——Apache Hadoop；
- 命名简单、大气的NoSQL数据库——Oracle NoSQL Database Enterprise Edition；
- ODI的Hadoop适配器——Oracle Data Integrator Application Adapter for Hadoop；
- 以Oracle“自家”数据格式、借助Hadoop装载并分析Oracle 11g数据的Oracle Loader for Hadoop；
- 针对数据仓库操作的企业级R语言；
- 另外，还有Oracle Enterprise Linux和Oracle Java HotSpot Virtual Machine作为基础的运行环境。

如果说以前的Oracle Exadata是战列舰的话，这次发布的Oracle Big Data Application一体机就是核动力航母。因为：

- 从市场定位看：在其他竞争对手还在大力推广“公有云”、“私有云”时，Oracle已经视图将这些“天高云淡”的内容装在一台机器中，允许企业根据业务需要，把它们一台一台摆在自己的计算中心里；
- 从产品组成看：Oracle所“融合”的不仅仅是自己独自开发的产品，更多是在借力开源社区的力量，通过“技术选秀”的方式，Oracle把其中的“好苗子”发掘出来、包装成具有明星大腕气质的Enterprise Edition；
- 从技术路线看：Oracle进行了一次自我破茧而出的过程，几乎完成了从关系数据库到NoSQL的跨界发展，而这应该是在对自己关系数据库市场和技术充分自信基础上的行为；
- 从用户使用看：Oracle Big Data Application中的每个产品如果单独由用户试用的话，相信都是个不小的挑战，更别说把它们“拧成一股绳”形成合力集成在一起。而且，这还不算完，为了能用好这组产品，需要开发、系统管理、运维各方面各层次都能跟得上，放眼全球能做到这点的企业比例寥寥。而Oracle用自己的最佳实践把它们打包在一起，通过管理员熟悉的同一个界面进行管理，对于用户而言其意义不言而喻——从“我恐怕不能”到“我能”。

作为新生事物，Oracle的这次尝试也有两点隐忧：

首先，Oracle的步子也许迈得比较大，而且对于能够在很大程度上影响Oracle产品采购的DBA阶层而言，他们要面临的新东西太多。Oracle Big Data Application中的很多内容又似乎在为其他部门加分，当技术升级遭遇职场政治时不知道会出现多少内生阻力。

其次，如果Oracle不能有效管理并引导用户需求，那么有可能用户某一天就会萌发这样的想法“既然NoSQL和Hadoop这么强大，我们为什么不直接try一下‘铁皮盒子’外面免费的NoSQL和Hadoop呢？”尽管对于Oracle的用户而言，捆绑了众多产品的Oracle Big Data Application俨然就是一位站立的泰坦巨神，但NoSQL们和Hadoop的背景也很深厚，它们更加不安分，按照现在的发展速度，谁难保证3~5年内它们不能登顶奥林匹斯山呢？



王翔
软件架构师，主要研究方向为XML、.NET、领域设计和PKI应用。工作之余喜爱旅游、写作和烹饪。

Google结构化 Web编程语言 Dart

编程语言的世界里，可谓江山代有才人出，Google推出的Dart究竟会表现如何呢？

GOTO大会（gotocon.com）是面向企业级软件开发筹办的全球性技术开发大会，参会者包括开发团队领袖、架构师、项目管理者、技术媒体等。GOTO大会组织者同时也是企业级软件的设计者和开发者，这也使得GOTO大会承载了来自技术型企业和开发社区的最新动态。

今年10月在丹麦举办的GOTO大会上，最吸引人的内容是Google新编程语言的首次亮相。在大会开场的主题演讲中，Google工程师Gilad Bracha和Lars Bak为开发者揭开了新语言Dart的面纱。

Dart语言面向结构化Web编程设计，是继Go语言后Google推出的第二门编程语言。Dart中文释义为“飞镖”，象征着这门语言灵巧高效的特性。Dart的设计目标是创建面向Web应用开发的结构化语言，用以创建灵活的网页程序。为便于开发者掌握并使用，Dart语法在设计上尽可能以开发者熟悉的形式出现。在浏览器兼容性方面，Dart考虑了各种浏览器差异，并在各浏览器平台上保持统一的性能与表现。

在门类众多的轻量级编程语言中，Dart具备不少吸引开发者使用的独特优势。首先Dart简化了Web应用的开发工作量；其次，Dart应用借助浏览器展示给用户，在具备良好平台独立性的同时，无需用户进行任何程序安装操作，并且对现代浏览器、移动设备、平板电脑的支持都表现良好。

Dart在定位上并非与JavaScript竞争，而是要填补Web编程方面缺少面向对象编程结构、静态类型、难以进行大型应用开发方面的空白，并为传统JavaScript编程带来大幅度的性能改善。

Dart代码运行阶段，可以依托浏览器内置的JavaScript引擎执行程序，同时Google鼓励各浏览器厂商在应用中绑定Dart虚拟机以实现Dart的原生支持。当然，Chrome浏览器已计划在未来版本中内置原生的Dart虚拟机运行时。根据性能测试数据，Dart虚拟机装载5万行Dart代码仅用时640毫秒，而从快照中加载相同的程序则更加高效，仅需用时60毫秒。

Dart开发工具支持上，Google基于Eclipse提供了轻量级的开发组件，用于编写代码并预览Dart应用。同时Google还正在开发一个名为Brightly的云端IDE提供给开发者，值得一提的是，这个IDE也是使用Dart语言开发实现的。

目前Dart语言站点（dartlang.org）上展示了语言规范、实例教程等内容。在Dart项目站点（dart.googlecode.com）提供了Dart类库、样例代码、Dart虚拟机及源代码、Dart转换JavaScript编译器等内容。

Google面向现代浏览器设计的Dart语言能够给Web编程带来良好的开发效率和优异的性能。随着浏览器应用规模化开发时代的到来，以及移动设备对浏览器应用需求的提升，我们期待新生的Dart语言为Web开发注入新的活力。



高昂
中国标准化研究院助理研究员，从事信息技术标准化研究工作。关注开源社区，也是OSGeo中国和InfoQ中文站成员。

智能卡在尴尬中演进

智能卡也出现了安全隐私泄漏的问题，该如何对待这种尴尬的事件？作为用户也许无可奈何。

本月的安全天下事注定要从智能卡讲起。近日，智能卡芯片制造商NXP Semiconductors（恩智浦半导体）对其Mifare DESFire MF3ICD40芯片的用户发出安全警告，表示这款芯片将在年底停止使用，并建议用户升级到Mifare DESFire EV1。这是对相关研究人员采用低成本设备破解了这款芯片的一种应对。此前不久，德国的研究人员基于边信道攻击的方法，使用价值约3000美元的设备，在七小时内获取了MF3ICD40使用的112位加密密钥。它的用户中

不乏敏感的部门，其中据说包括NASA。再往前回顾，2008年，该公司上一代芯片Mifare Classic也是被研究者破解，促使大量用户转移到了MF3ICD40之上。

在国内，两个重要的桌面安全企业奇虎360和金山都在本月与公交IC卡这一智能卡的重要应用产生了关联。

首先是媒体曝出奇虎360的两名工程师因破解北京公交IC卡并进行充值而被法办的新闻。根据360的官方声明，两名工程师参与了公交卡的安全性研究，由于“法律意识淡薄，没有能够抵制住诱惑”而导致了此事的发生。360已经将两人除名。根据报道，法院分别判处了两人拘役的处罚（其中一人缓刑）。

而后，金山指出北京公交IC卡服务网站存在漏洞，凭借卡号即可查询持卡人的刷卡轨迹，这有可能暴露用户的位置信息和出行习惯等。这一事件引起了各方的争议，有一些安全研究者认为这不是一个很严重的问题。首先卡号只能用于查询、不能用于消费；公交卡也并非实名制卡，其定位信息不是实时的，因此其对隐私威胁的程度是轻微的。正如网友所说，无非是“给多疑的妻子增加了一种对老公查岗的方法而已”。站在个体安全的角度，我也倾向这种观点。而且对于公交卡海量发放的庞大基数来说，如果建立面向密码的安全机制，必然带来密码分发、管理和废止的机制，本身也要带来一定的成本支出。

然而，随后有应急工程师在某企业网出口上发现了批量获取公交卡信息的流量，这就提出了本问题的一个新的视角。从公共信息查询环节来说，批量地获取相关数据，显然可以从中进行时间和站点的客流分布等挖掘、可以进行群体性充值和消费习惯的研究。

如果我们把公交IC卡的卡号与查询服务进行一个原则抽象的话，可以看到它确实不是一个安全的体系：

- 其卡号是顺序流水号，而不是符合某种认证规律的编码，即只要知道其中一个有效卡号，就能轻易地批量产生其他有效卡号；
- 其卡号是类似查询的唯一鉴权凭证，没有其他扩展的保护信息，例如信用卡的有效期、CVV/CVC验证码等；
- 开放的查询接口对获取相关信息的行为未做其他限制，如单一IP查询次数限制、图片验证码机制等，这种设计必然会出现有人尝试批量获取数据。

有网友提出建议，说可以凭借卡号建立初始查询密码。这肯定不是一个合适的选择，由于卡号本身是流水号，势必会有恶意攻击者对大量卡号批量修改密码。如果希望由用户自行设置密码，在没有初始分发机制的情况下，目前相对可靠的方法是凭借IC卡到指定的自助终端或者充值网点去设置。

智能卡在不断演进，智能手机系统也是如此。Google和NSA正在联合研发基于Android 3.0的安全内核。目前仍不清楚这是一个独立的系统，还是一个效仿SELinux模式的补丁。但随着IT寡头格局的演变，我们将逐渐看到强大的产业能力对美国国家安全战略的支撑能力。每每思及此，我都会想起一位老师对我的教导：“沦为用户，即意味着承认着被主宰的幸福。”



肖新光
网名江海客，安天实验室首席技术架构师，研究方向为反病毒和计算机犯罪取证等。微博：weibo.com/seak。

淘宝商城事件的影响

淘宝商城事件让人惊觉电子商务由来已久的弊病，电子商务该怎么做，本文详细解说。

在中国电子商务发展史上，10月的淘宝商城事件，绝对称得上是一个重要的标志性事件。淘宝商城近日将以前商户每年缴纳的6000元技术服务费提高至3万元和6万元两个档次。另外，根据所经营或者代理的品牌类别不同，2012年的商城诚信保证金从现行的1万元，调整至1万元至15万元不等。这项新规立即遭到了中小卖家的强烈反对和抗议。数千中小商家采取“网络暴力”对淘宝商城的大卖家进行恶意下单，迫使这些大卖家的大部分商品下架。甚至国家商务部也介入了淘宝商城事件，发表了自己的看法。我们这里，暂不讨论这个事件的是是非非、谁对谁错，是谁又一次破坏了契约精神，单就这个事件本身，对今后电子商务的影响，做一下自己的展望。

首先，我认为，在中国，电子商务的下一波浪潮应该是由传统企业推动，是现有B2B2C平台的延伸。如果没有

这次的淘宝商城事件，顺理成章的，一个传统企业想从传统卖场走向自己的电子商务之路，那么最好的步骤就是首先在淘宝商城上卖卖试试，等有了稳定的客户群、有了电商的感觉，再在京东、当当上开店，在新浪上织微博，做进一步的市场推广。再然后是否要有自己的B2C独立网上商城，可以根据自己的实力和基因，再做进一步的判断。然而，本次的淘宝商城事件，很可能会打乱部分传统企业对在淘宝商城上开店的看法，大大加深平台和自身之间的彼此不信任感。有了这层的不信任感，就会留给新的B2B2C平台加入的机会，从而造成竞争的加剧，打破淘宝一家独大的局面，在B2B2C平台形成几个类似于B2C京东、当当商城这样的可以与淘宝抗衡的平台。

其次，对于想转入淘宝商城的C2C平台的大卖家来说，他们可能会延缓或者直接选择不再加入淘宝商城。这些大C与淘宝之间，不信任感也会大大增强。不会有人再天真地相信，淘宝就是自己终身可以奋斗的平台，从而导入淘宝站外流量的意愿会大大增强，而不再专心研究淘宝自身的流量和直通车、打造爆款的技巧，只是把淘宝看做一个出货的平台，内心的那种对淘宝平台的草根之爱慢慢趋于平淡。那么在百度上做广告的卖家就会大大增加，百度搜索是这次事件的最直接的受益者。由于百度不能直接搜索到淘宝的商品信息，所以淘宝客、论坛和其他免费信息发布平台又有了一些新的机会。

第三，竞争的加剧和相对机会的增多，会促使小的B2B和B2C网站开放自己的数据，从而形成一个相对比较大的虚拟的B2B2C平台，这时留给第三方软件开发者的机会就比较多了起来，这些缝隙，总是需要一个或者多个平台去填补的。如果这些缝隙，再跟社交网络结合起来，那么一个全新的类似Facebook的商业模式，也许就会诞生。最有可能的突破口，就是在新浪微博上开店的商城。所以新浪微博，无论是社会新闻，还是微博商城，新浪都是此次事件的最大受益者。比较大的传统卖家从最先选择淘宝商城做尝试，很可能会选择从自建B2C平台，从而在新浪微博上开始做尝试。新浪如果此时选择自己做一个B2B2C平台，那么它的影响将是爆炸性的，会对淘宝构成最大和最直接的打击。



邢波涛
北京新软孚信息技术有限公司技术负责人。关注SaaS管理软件和B2B、B2C电子商务的融合。

业界报道

云端魅力

2011年10月11日，微软技术大会TechEd 2011在北京万事达中心隆重开幕。作为IT业界最富有影响力的年度盛会之一，TechEd主题演讲以雷霆万钧之势展开，微软新产品激动人心的魅力得以完美呈现。置身其中，这一切让人充分感受到了微软在云计算领域的努力与执著，也坚定了我们在云计算道路上的决心。



微软大中华区副总裁、首席云计算战略官谢恩伟深刻阐述了微软云计算的“云+端”思想，并通过诸多案例讲解了微软的云计算战略平台。

微软的云计算战略极其灵活，而又富有传承性。企业既可以使用Windows Server和SQL Server这样的传统计算平台构建私有云，又可以根据情况过渡到以Windows Azure和SQL Azure为代表的新一代云计算战略平台，从而大幅提高企业生产力。

谢恩伟邀请嘉宾做了一系列精彩的演示，其中最让人期待的两颗明星莫过于开发之中的Windows 8和即将面世的Windows Phone。Windows 8重新设计的开始界面让人眼前一亮，它内置了应用程序商店，占用资源更少，并支持ARM芯片，适用于各种PC和平板电脑。我们可以登录http://dev.windows.com下载最新的开发预览版。

而在作为微软云计算战略重点的移动领域，微软结盟诺基亚，即将推出Windows Phone。Windows Phone以人为本，绚丽的演示让在场的观众目不暇接，不时响起热烈的掌声。观看至此，不由得令人感叹，移动领域必将烽烟再起。

一位消费电子创业者的Smart TV创业观

记者 / 闫辉

精美的Android系统机顶盒，再配上独特设计的无线电容触控装置，接上普通电视，就可以变成支持各种App应用的Smart TV（智能电视）。这也是移动平台进入家电领域的尝试。Android系统已经改变了智能手机市场，会决定和统一未来的智能家电领域吗？让我们与一位资深IT消费电子创业者交流，共同探讨移动平台和智能家电的创业前景。



宫润玉：翰骏电子科技创办人。15年消费电子产品设计经验，6年摩托罗拉半导体产品经验。曾任职于Java虚拟机服务厂商 Esmertec AG，并创建台湾地区分公司；掌亚国际科技（韩国三星IC设计服务独家合作伙伴），并创建大中华区产品开发团队。

Smart TV是一个开放的系统

《程序员》：请简单介绍一下你的个人经历和做Smart TV创业的背景？

宫润玉：我是韩国出生的华侨，在首尔读书到高中毕业，大学去了台大资讯工程系所，研究所毕业后就留在台湾，大部分时间都以台湾为中心，直到五年前来到上海。

我在摩托罗拉半导体服务期间，负责PowerPC（当时苹果的处理器）在通信领域的应用。1996年深入到半导体行业。2002年开始做手机，任广达的产品经理，当时我们的客户有松下和西门子。后来我去了Java虚拟机厂商Esmertec AG，建立台湾业务。

2006年起，被外派到上海，为三星半导体的处理器建立技术支持团队。因此，我的经历主要与半导体处理器和操作系统相关，从嵌入式Linux、Windows CE到Android。

经过很长时间的观察，我从2009年开始创业。因为在手机产业做了近十年，所以我认为手机行业现在是规模市场经济，并不是一个好的创业题材。而电视在互联网应用方向相对落后，反而是个不错的方向。

当时，我们二十多人的团队主要做三星处



下一代的遥控会充分利用多点电容触摸

理器的解决方案，从处理器的发展来看，手机的处理器和电脑处理器的能力越来越接近，特性也越来越雷同。PC处理器的领域主要是家庭和办公室，而手机处理器可以用来拓展新兴应用，尤其是在家电市场。刚开始在三星公司内部创业时，这个项目叫做PCTV。后来，我们将这个项目变成了独立的公司。

《程序员》：你做Smart TV的设计理念是什么？

宫润玉：做产品，我们特别强调大众消费者的声音。这个产品要很小、易于掌控、看得到、摸得着、可以评论其产品设计、可以替换和选择。

在我的定义中，Smart TV也是一个开放的系统。操作系统和应用市场都是开放的。我们要使用现有的资源和应用，用最经济的手段把智能电视做出来。有了这些认识，选择Android平台就顺理成章了。

同时，iPhone让触控来势汹汹，立刻改变了人们的使用习惯，这种方式在未来也会继续延续下去。触控不仅影响了操作系统，同时也影响了应用开发。因此，我们对触控操作模式非常重视。

现在电视上主流的交互设备是遥控器。遥控器有方便使用的一面，而一旦进入到一些比较细微的操作，就力不从心了。所以，我们认为下一代的遥控会充分利用多点电容触摸。欣慰的是，触控的技术和芯片也在飞速发展，解决方案也比较成熟。

站在终端产品设计者的角度，上游产品和技术的不断提升，可以让我们更好地应用到设计理念中。整个产业本身的发展就是这样，我们在做一个顺势推舟的产品。

Android 系统在Smart TV 上优势明显

《程序员》：根据你的经验，用Android系统来做Smart TV，有哪些优势和不足？

宫润玉：既然我们选择了Android，优势显而易见：第一是开源，第二是简单易用。

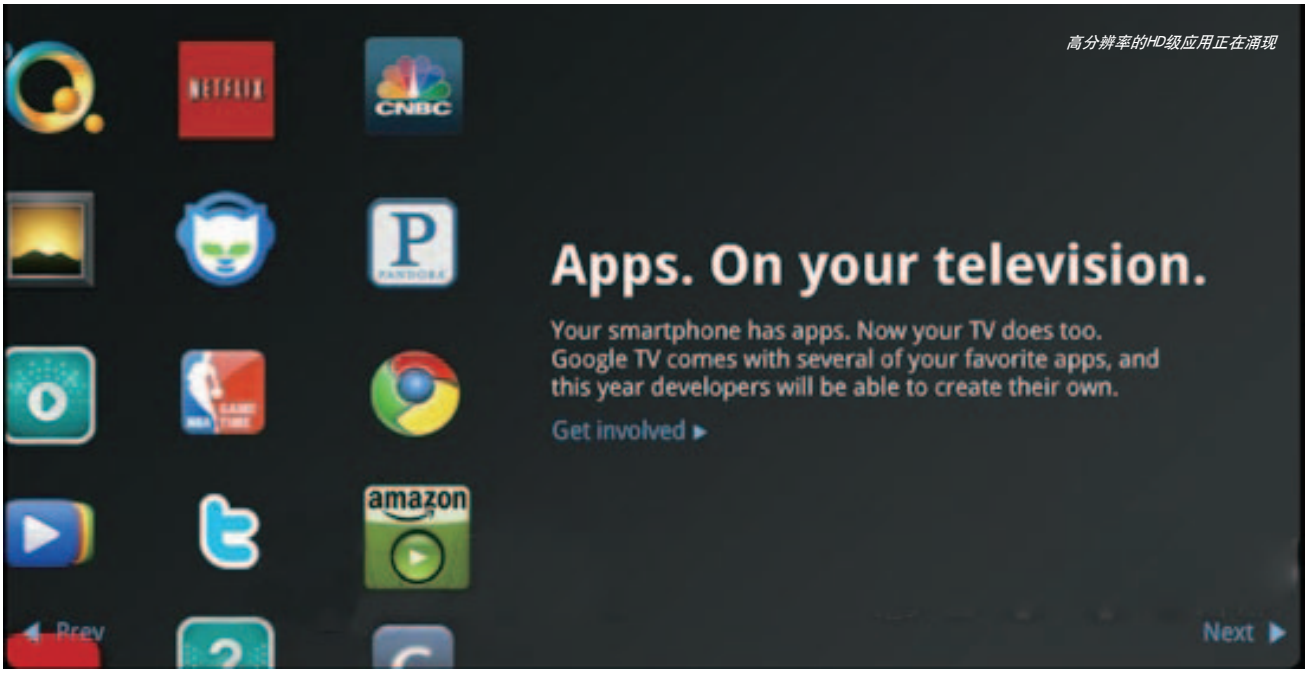
我们是2009年开始研发Smart TV，那时刚发布Android 1.6版本。这个版本还不支持诸如高分辨率、高清格式、网卡驱动和鼠标键盘等性能。我们的第一代产品还是用Windows CE加三星手机用的芯片实现。

直到Android 2.1版本发布后，才开始慢慢完善这些支持。在平板出现后，状况就有很大改观。因为平板的功能需求不同于手机，芯片平台商逐渐将这些缺点弥补了。

更重要的是应用。在2010年，Android的高分辨率应用还很少。这样的应用在电视上的视听效果比较粗糙。也是随着平板的流行，到2011年3~4月，我们发现开始出现很多高分辨率的HD级的应用。比如视频网站的HD应用，使得在电视上看电视剧和电影变得非常棒。

这些变化让我感到十分欣慰，说明我们的方向走对了，和平板的价值曲线是一致的。

《程序员》：三星和国内的一些电视厂商都在做智能电



高分辨率的HD级应用正在涌现

视，你们为什么没有选择直接与电视厂商合作，而是做外置的独立式解决方案呢？

宫润玉：每个产业本身都有它独特的文化。我曾经和创维、TCL、三星都有接触。应该说，电视产业信息化的发展相对比较落后，是一个比较封闭的产业。所以，这些电视厂商更希望把一个概念从头做到尾，在新的功能和技术革新出现时，更愿意与传统的功能一起做整体考虑。

举个例子，摇控器看起来简单，但有了20~30年历史。每个公司都有自己的想法，但作为互联网应用，文字输入需要很多创新，原来的遥控器设计很难承担这样的需求。

另外，和电视厂商合作，我们只能作为一个供应商，变成电视的一部分，从功能呈现到线路板的空间都受很多方面的制约。而各厂商负责具体事务的人可能对智能电视未来消费者的习惯并不了解，也不会特别重视。

现在的电视行业，需要不一样的商业模式去刺激它。这是我们的观点。

中国Smart TV产业需要工业标准

《程序员》：如何看自己在整个产业链中的角色？

宫润玉：我们在做产品模式时一直提醒自己，我们在整个产业链中的角色是什么？价值在哪里？生产、渠道还是产品设计？

好产品一定要更接近市场端，做一件事情最难决定的就是放弃什么。事实上，我们将机器的生产尽量外包出去，而将产品的研发抓在手中。

我们最有价值的核心有两点。

第一个是交互界面的设计。小到游标的形状，大到遥控器本身，如何做出创新，我们会百分之百地投入。现在的电视交互，一些是利用遥控器，一些是通过体感识别，触控是我们选择的一条路径。

第二个是应用软件的合作。我们和一些大的做内容和电视游戏相关的开发商有一些合作模式。

我们的Smart TV是一个载体，能够载入各种不同的内容。手机和PC平台的应用移植到电视上，还需要做一些转换，体现出电视媒体的特性。我们会沿着这条路去走下去。

《程序员》：对于移动应用开发者来说，你认为为Smart TV开发应用需要注意哪些特殊的关键点？

宫润玉：首先，电视的屏幕很大，屏幕对应用的画面要求很高。其次，Smart TV的带宽足够大。在做应用时，要尽量利用这些优势，在产品设计部分要有足够的投入。

此外，Smart TV的人机界面没有统一的标准，对开发者来说这是一件令人头疼的事情。现在做Smart TV的厂商之间基本上是没有交流的，没有产业联盟。

这样，当没有形成规模时，就很难吸引更好的内容和应用。中国的Smart TV产业要考虑如何拟定出一个工业标准，让这个市场能够得到更加标准化的发展。

普适计算20年

记者 / 付江

今年是普适计算领域奠基性论文*The Computer for the 21st Century*发布20周年，业内顶级的国际会议——第13届ACM普适计算国际会议UbiComp 2011也于9月底首次登陆中国。

回眼望去，普适计算最早起源于1988年Xerox（施乐）PARC实验室的一系列研究计划。在计划中Mark Weiser首次提到了普适计算的概念。1991年Mark Weiser在*Scientific American*上发表了文章*The Computer for the 21st Century*，正式提出了普适计算（Ubiquitous Computing）。

云计算和物联网是实现普适计算的基础

普适计算认为计算能力将融入网络，融入各种环境和人们的生活。为此，计算机会更小、更廉价，有网络连接，将超越图形界面，可以与环境和人有更多的交互手段。Mark Weiser还为这种智能设备定义了三种形式：Tabs，尺寸以厘米计算，可穿戴；Pads，尺寸以分米计算，手持；Boards，尺寸以米计算的大型设备。

20年前“普适计算”能力的定义，甚至已经成为一些应用开发者的创业思路。移动开发者围绕着人、环境、智能设备来提问题：智能设备能采集什么环境信息？能整合哪些计算能力和网络资源？为哪些人？解决什么问题？利用智能设备强大的环境感知能力和多样的交互手段，不少应用开发者给出了答案，各类LBS应用就是个很好的例子。



James Landay认为普适计算的最大挑战是利用技术解决现实生活中的问题

众多的国际IT巨头例如微软、苹果、IBM、英特尔、谷歌……都是普适计算的热衷推动者。从某种程度来看，普适计算的概念也为时下火热的云计算、物联网、移动计算勾画出了一幅蓝图。

对此，UbiComp 2011联合主席、同时也是微软亚洲研究院访问学者、华盛顿大学计算机科学与工程系教授的James Landay（刘哲明）认为，云计算可以看成是基础设施和平台，物联网强调的则是网络互联，却没有清晰说明联网之后做什么。它们都是实现普适计算的一部分。而移动计算与普适计算在虚拟化、智能家居、智能教室等众多领域有着广泛的应用场景，但同时也面临着无线通信技术、移动设备、移动性能等带来的诸多挑战，需要从设备、网络、操作系统、数据库、中间件、数据挖掘等方面给出解决方案。

在他看来，20年来，普适计算往前走了很多，在理论基础、硬件设备、应用软件上都取得了很多成绩，但仍然有很长的路要走。

最大挑战并非来自技术

今年普适计算大会的核心议题是用普适计算技术来解决日常生活中的问题，例如环境保护、健康问题、人类教育等方面。James Landay认为，普适计算研究领域遇到的挑战有3个方面。最大挑战并非技术本身，而在于如何利用技术真正解决现实生活中的实际问题。其次是如何跨领域、跨文化、因地制宜地解决当地的问题。此外，工业界和学术界的交流也尤其重要。

普适计算的成果越来越深入影响到普通人的生活，传统意义上的计算机正在消失，而且，人们不会注意到它的离开。同“少即是多”相反，在普适计算领域内，多即是少，也就是说，如果天时地利人和都实现，用户之间需要使用的技术更少，需要处理的任务也更少。Mark Weiser 20年前的论文准确预测到了今天，那么，未来20年又会怎样呢？

IBM王小虎：架构师不是一日炼成的

文 / 付江

本文是“架构师领袖”系列报道的第三篇，在前两篇中，IBM软件集团大中华区合作伙伴技术支持总经理王小虎分别从《重视架构方法论的紧迫性》以及《架构师面临的十八般武器》的角度，阐述了企业架构发展趋势、挑战、解决之道。本篇则将分享王小虎如何从技术人员走向架构方法论业界领袖的历程和心得，与广大技术人员共勉。

博览群书受益终生

《程序员》：您儿时的梦想和爱好是什么？这些爱好对您以后的职业发展有什么影响？

王小虎：坦率地说，我小时候并没有什么特别执著的梦想。父母也没有“逼”我上各种兴趣班，而是给了我一个很宽松和自由的成长环境。现在回想起来，儿时的习惯里受益终生的就是特别喜欢读书。大量阅读不仅培养了我的求知欲，扩大了知识面，也为日后工作中快速阅读能力的提高打下了扎实的基础。小时候读的书不限题材，记得我在上小学时就已经把中国四大名著通读了。

《程序员》：您什么时候开始接触到计算机？当时什么感觉？

王小虎：我大学报考的就是计算机系。之所以选择这个专业，是因为当时觉得计算机很神秘，高科技的东西给我很炫的感觉，我觉得自己可以适合做这行。当初的志愿是我自己填的，父母并没有干涉我的选择。大一时，我第一次正式接触到计算机，学的第一门语言是Pascal，一下子就觉得异常兴奋，我立马就爱上了这个聪明的“家伙”，我当时操作计算机做了一些有趣的事情。现在看来，虽然当时做的只是一些简单的运算处理，但可以把我的想法通过编写程序输出的方式来实现，得到想要的结果，这让我觉得非常有趣。

《程序员》：您毕业后找工作的情况怎样？

王小虎：硕士快毕业时，导师问我，要不要留校。我想了想，学校的氛围也蛮好的，于是就留下了，在北航做了两年教师。随着时间的流逝，我发现自己最擅长的不是常年在学校里做理论研究，而是更偏爱将理论和技术结合起来，应用到具体的企业项目中。这时，我开始考虑找一家企业了。当时选择的主要是跨国知名外企，后来通过面试加入了IBM，这一待就是13年，一直到现在。

职业生涯的两次转型

《程序员》：刚加入IBM时是什么感觉？

王小虎：第一个职位是数据库工程师。回头看来，当时很是迷惘。首先IBM公司很大，我要花一些时间去搞清楚各方面的关系。其次，我的角色已经转变了，从过去的研究人员，转到了销售类的技术支持工作。这个过程其实是需要一些时间去体会的。因为不仅仅是做技术这么简单，过去只要把自己手头的程序写好，检测出Bug，调试通过就行了。而作为销售的技术支持不仅需要更多的和团队配合，还要有很强的销售意识和沟通能力，这些软技能都是需要时间来积累的。我用了近一年的时间使自己变得可以从容地处理这些事情。

《程序员》：如果从您加入IBM到现在，划分几个阶段，您会怎样分？

王小虎：大致可以分为四个阶段吧。第一个阶段就是头一年，从各种迷惘到逐渐比较自如地去工作，完成从研究人员到技术销售的转型，在这个过程中逐步加深了对公司产品和解

决方案的研究和理解。

第二个阶段是从1999年~2001年，在项目中我让自己的理论和实践能力都更加扎实，可以去应付一些很复杂的项目和有挑战性的客户。

从2001年~2007年我开始第二次职业转型，从技术个体转向IBM的一线经理（管理岗位）。开始更加关注如何提高团队战斗力和管理能力。

从2007年起，我开始领导IBM软件集团大中国区的合作伙伴技术支持工作（下面有经理帮我带团队）。重要的是我学着去更多地授权和培养经理让他们更好的建设和管理团队。此外，更多地考虑如何通过一些创造性的计划和措施来促进业务的发展，然后带领团队去实现它并且大家都从中感受到乐趣。

架构师不是一日炼成的

《程序员》：很多技术个体正走在转型为领导者的路上，您有没有一些建议给他们呢？

王小虎：最重要的就是找个导师。技术人员在转型时，对某些事情的处理会缺乏经验，很多时候并不是黑或者白这么简单的事，如果有个导师指教会有很大帮助。其次，需要花更多时间去当一个教练的角色。这时，你个人做得好不好，往往并不能直接让整个团队的成就就好。或许你过去是个技术高手，但这时候你已经很少有时间去做那些非常细节的事情，所以就需要花时间去当一个教练的角色，去鼓励团队成员取得更大的成就。

《程序员》：的确，现在很多技术人员都在低头写代码，其中有部分人打算转型到架构师方向，您觉得他们在这种转型中有哪些注意事项呢？

王小虎：首先从工作角色上来说，会有一个过程。不是说今天去学了很多东西，明天就能变成一个架构师，而是需要沉下心来，低头能写程序，抬头能把握技术方向，在工作中不断学习、总结、应用这些方法论。比如说，你现在是一个程序员，想成为一个架构师，就应该多参考时下比较流行的方法论，先在小团队里应用起来，再逐步推广到大的项目中。



王小虎：程序员转型到架构师，需要沉下心来，低头能写程序，抬头能把握技术方向，在工作中不断学习、总结、应用这些方法论。

就架构师而言，还需要有及时了解新技术的兴趣和激情，并且能够针对新技术进行分析比较，知道该技术的优缺点，才能作出合理的架构决策。对自己的团队有着充分的了解，知道团队成员的技术能力与潜能，了解他们的秉性，找到合适的人完成合适的技术任务。很重要的一点，架构师要具有比较深厚的行业经验，能够对架构和技术的发展有一定的预测性。

实践架构方法论是领导工程

《程序员》：您觉得一个企业实施好架构方法论的关键是什么？

王小虎：首先，公司的领导要重视架构，从上往下推广会容易些。其次，在公司内需要一个培训和教育的过程，从组织的角度来说，应该有一些类似于架构委员会的组织，来负责架构管理、流程、推广等诸如此类的事情。再次，需要有一个可持续发展的规划。通过这些过程，逐步实践到具体的项目中。公司的架构师需要熟悉架构方法论，对行业了解，然后能针对具体项目，对方法论进行适当的裁剪，运用到项目解决方案中。只要坚持下去，企业很快就会感受到架构方法论带来的巨大好处。

中国式产品管理，



探索中前行

文 / 马博

1998年，宝洁公司的EPS（Earnings Per Share，每股收益）只有14%~15%，低于华尔街的期望。

1994~1999年，宝洁公司的收益始终在1.4%~5.5%之间，低于公司内部7%的目标。

在1998年前的整整十年中，宝洁公司始终没有推出一个有影响力的新品牌。

究其原因，除了受到更多传统竞争对手（如联合利华）以及新兴竞争对手（如沃尔玛自有品牌）的冲击外，还有一个更为重要的原因是宝洁曾经引以自豪的品牌管理系统出现了创新匮乏、效率低下的问题。

于是在1998年9月，宝洁公司宣布了一项名为“Organization 2005”的公司重组计划，重点是基于品牌和产品线重构公司的组织架构、工作流程、企业文化和薪资结构，其目的在于增强公司的竞争优势和营业效率。

“Organization 2005”计划实施后，宝洁公司的年销售额增长了6%~8%，年收入增长了13%~15%。

我们知道，产品管理诞生于宝洁，但通过这个案例可以看出，在宝洁自身的产品管理发展的80年中，其组织结构的形式也随着企业自身的发展和市场的变化发生了巨大改变。

我无意去赞美宝洁的伟大，只是希望通过宝洁的“Organization 2005”计划来告诉中国正在或者意欲实施产品管理的企业以及期望或者正在从事这项工作的个人，无论是对于国外企业还是国内企业，产品管理的具体形式不可能一成不变，必然会受到具体时代的影响而处于一种动态的调整之中；唯一不变的只是产品管理的核心思想：让企业将有限的资源投入到最有价值的产品上去。

但反观国内企业，对于产品管理，一切都不是那么尽如人意，国内企业似乎对类似产品管理这样的舶来品始终有着一种莫名的纠结，而这种纠结造成的直接后果就是用浮躁的心态理解和实践着它们。

企业的三种心态

总体说来，国内企业对于产品管理有着三种在我看来很不是滋味的心态。

跑偏的

凡是舶来品，在国内总是有流行的机会的。

现在产品管理也算是逐渐开始流行了，各个行业都有不少企业开始实施产品管理，但有不少企业却实施了很多名不副实的产品管理，例如把产品管理部门作为市场或者研发这些部门的二级部门。当然，这种结构是靠谱的，但干的活却并不靠谱。很多产品部门基本沦落为这些部门的“碎催”，凡是上级部门不想做、懒得做的活就交给产品管理部门来做。

产生这个问题的原因有以下两种。

- 在实施过程中因为企业内已有体系和部门的阻碍，使产品管理的实施逐渐偏离了正确的方向。
- 企业在实施产品管理之前，就把产品管理理解成这样。

因此，我把这种情况叫做“跑偏的”。

跟风的

说到底，企业就是由两类实务构成的：管理和业务。

我有一个观点：在管理上，应该具有中年人的心态；在业务上，则应该具有年轻人的心态。

管理像中年人，说明企业具有独立思考的能力，在应该使用什么样的管理上会做到实事求是；而在业务上，则应保持一种年轻人的激情和冲劲，不惧怕任何挑战，敢于用新的思路来尝试各类挑战。

但现在很多中国的企业正好做反了，在管理上像年轻人，在业务上像中年人。

对于流行的管理思想，尤其是舶来品，也不管是否适合自己或者是否做过实事求是的评估，只要是媒体鼓吹的、只要是有大企业实践过的，就拿来用。结果很明显，由于这个原因失败的案例太多太多。

对于产品管理也是如此。很多企业连产品管理到底是什么、到底能为企业带来什么、针对产品管理企业是否需要做一些必要的调整、如果实施产品管理对企业来说会有什么风险、应该如何实做事求是地实施产品管理等等，都没有进行清晰的评估，就开始用一种大无畏的精神实施，而原因仅仅是因为区区“流行”二字。对于此，我只能说八个字：精神可嘉，效果不好。

如果能把这种精神对调一下，那该多好！

救命的

企业对于其他管理思想是否是这样的态度，我不知道，但我知道有一些企业就是用这种心态来看待和实施产品管理的。

产品管理不是灵丹妙药，无论得什么病吃了都会好；产品管理也不是万能胶，贴到哪里都起效。产品管理有其自身的特点和适用性，总结了一下，产品管理事实上在八个方面会给企业带来必要和显著的价值提升。

- 企业长期利益和短期利益的更好结合。
- 产品、产品线、产品组合之间的关系更加紧密和协调。
- 更为明确的产品负责制并贯穿始终。
- 团队业务效率明显提升。
- 企业内部隐性成本明显降低。
- 用户满意度明显提升。
- 产品方向更加明确，目标用户更加清晰。
- 产品周期管理更合理、更顺畅。

现在，从国家到企业都在提产业升级、产业转型，这是从宏观上为产品管理的进一步发展奠定了基础，但对于企业来说，第一步并不是考虑怎样去实施，而是实事求是地评估自己是否真的适合实施产品管理。如果适合，应该采用怎样的构架形式。

在国内，有很多企业具备了实施产品管理的条件。因此，是否实施产品管理不是我们重点探讨的问题，我们更希望探讨的是什么时候实施产品管理。

必须承认，实施产品管理是企业应对更加复杂的市场环境的必然之路，因为中国的市场化会越来越强，而产品管理就是一种完全市场化的产物。

但我们还必须看到，在有些企业适合实施产品管理的同时，也有很多企业并不适合。因此，千万不要把产品管理当成是救命稻草。尤其是对于那些有问题、但并不适合实施产品管理的企业来说更是如此。

我的建议是要把产品管理当成发动机，在企业具备升级转型的基础上让产品管理推进一步。

产品经理需注意的三个问题

如果说企业的问题是平台和环境的问题，是会影响产品管理实施效果的因素，那么个人的问题就是素养和技能的问题，是会影响产品管理的价值实现的因素。

同样，个人在产品管理的操作层面，也有三类需要引起从业者注意的问题。

知道做什么，但不知道怎样做

这类产品经理有三个特点。

- 对行业和产品有深刻的认识，能够抓住本质和关键点。
- 能够提出有市场价值的产品想法。

■ 缺乏系统、规范的操作能力和产品管理工作思路。

这种产品经理就是典型的“理论强，实战弱”，他们能够很好地理解高层的意思，但实际情况是产品经理不仅是一个会“想”的角色，还得是一个会“做”的角色，许多关键的业务必须靠自己亲力亲为才可以。

我们都熟悉一句话“产品经理是没有权力的小CEO”。从这句话可以看出，本质区别可能就在这里：CEO只需要会想就可以了，而产品经理不但得会想，还得会做，等什么时候不用做了，那就成为真的CEO了。不过在成为真的CEO之前，还是需要先把该做的做好才行。

不知道做什么，知道怎样做

这类产品经理有下面两个特点。

■ 缺乏主动思考的意识和能力，通常只是被动地做一些执行的工作。

■ 执行能力很强，能够运用各类工具方法模型。

这类产品经理和第一类正好相反，会做不会想。做个类比，这类产品经理可以成为军队里攻城拔寨、奋勇杀敌的将军，但却很难成为运筹帷幄、统观大局的元帅。他们只会执行拿下山头的命令，但做不到因地制宜地考虑怎样去用最合适的方案拿下这个山头。

之所以诞生产品管理，有一个很重要的原因就是因为企业内的产品太多了，作为高层已经无暇顾及所有的产品。尤其是在市场越来越复杂的情况下，设立产品经理，对于高层而言，作用就在于能够扮演高层智囊的角色，替高层分担这些产品所带来的压力，并提出产品发展的长期合理的建议。

但这种类型的产品经理怎样能够很好地做到这一点呢？

因此，我们看到的情况是，虽然有了产品管理、设立了产品经理，但高层依然很累，依然在许多事情上得亲历亲为。有些企业高层甚至感慨道：“说真的，在没这些产品经理之前，我都没这么累过。”

这是产品管理的问题吗？自然不是，很大程度上是产品经理的问题，也就是不知道做什么。产品经理本来应该是高层的克隆，但克隆没成功，只有四肢，没有大脑。

不知道做什么，不知道怎样做

既然两个“不知道”都占了，这种类型的产品经理也就不必多解释了。

在探索中前行

不过倒也不必为这些太过失落，因为对于一个进入中国才十来年、快速发展不超过六年的管理思想而言，出现这样的问题也在意料之中，我们唯一要做的就是对这些问题进行分析并找到合适的解决方法。

产品管理对于中国企业来说，既老又新，说它老，是因为它的诞生要比现在流行的很多管理思想要早得多；说它新，是因为不但在国内，即使是在世界范围内引起高度关注也是从20世纪80年代末~90年代初才开始的，尤其对于中国企业来说，更是如此。

或者可以这样说，在这个管理思想的认识和实践上，我们并没有被国外同行落下太多，我们现在需要做的就是思考如何才能把本来不大的差距继续缩小，直到因地制宜、因时制宜地让产品管理思想本土化、体系化和职业化。

再回到开篇宝洁的案例上，虽然我无意赞美宝洁的伟大，但在最后，我还是不由地要为宝洁审时度势的改造大声叫好，这或许应该成为中国企业实施产品管理的一个标杆。当然，这个标杆不是建立在“Organization 2005”这个重组计划上，而是建立在宝洁为追求产品管理最合适的实现形式上的不断探索和实践的勇气和耐心中。

中国的产品管理，其实才刚刚上路，我们依然需要在探索中前行。

这一切才刚刚开始！



马博

UCPM（中国产品经理联盟）发起人之一。现主要从事产品管理的研究与实践。曾经涉足重工、广告、会展、互联网、教育等多行业领域。

【编者按】一个体系的运转，除了需要合理科学的架构，从根本上说，还得靠有高素质的相关人员来支撑。具体到产品管理体系中，最直接的体系支撑人员就是产品经理，而现实的情况是，许多企业设立了产品管理体系，但却因为缺乏合格的产品经理而使体系的效用发挥大打折扣。这其中蕴涵的问题是：产品经理应该具备什么素质和技能，才能担负起支撑产品管理体系的运转？UCPM（中国产品经理联盟）通过对上千名各行业产品经理的调研和分析，总结出一个产品经理人才模型，如图1所示。在这个人才模型中，产品经理被要求具有三个方面的素质：个人素养、管理知识和核心能力。接下来，我们就通过《产品经理的“巧实力”：个人素养》、《产品经理的“软实力”：管理知识》和《产品经理的“竞争力”：核心能力》三篇文章对这三个方面进行详细的说明。

产品经理的“巧实力”：个人素养

由于产品管理工作的特殊性，因此，在所有采用了产品管理体系的企业中，无一例外地对产品经理这个具体的“人”有了要比其他岗位更为严格的素质和修养方面的要求。具体来说，即为图1中的个人修养、创新意识、沟通能力、抗压能力、自管能力五个方面。

这是作为一个产品经理所具备的基本素质，如果不具备这些基本素质，即使其他能力再好，也无法成为一名优秀的产品经理。

因此，我们把个人素养称为产品经理的“巧实力”。

个人修养

个人修养的好坏是一个人成熟与否的表现，修养的最高境界，应该是“宠辱不惊”。

产品经理的工作具有两个特点：一是工作的“权责不对等”；二是“必须依靠团队”开展工作。这种看起来是自相矛盾的定位，就决定了产品经理必须具有非常好的个人修养才可以。

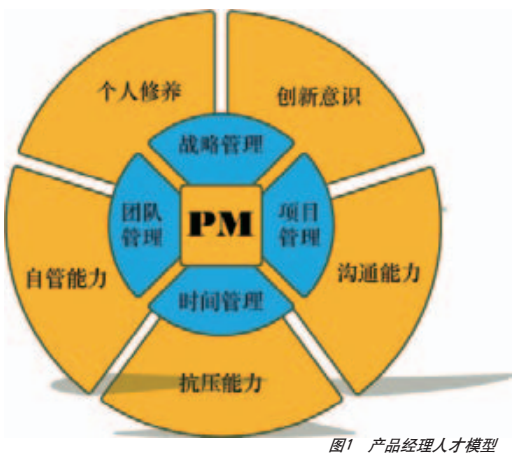


图1 产品经理人才模型

例如，当研发团队因为某些原因对你提出的要求敷衍或者拖沓时，你该怎样做呢？是质问、投诉，还是愤怒？

我们相信，当你遇到这种情况时，如果你不是产品经理，做出以上任何反应都是很正常的；而一旦作为产品经理，如果你做出这些反应，那么不但对你的工作无益，反而会影响你在产品团队中的威信和声望，甚至是职业前景。

有朋友曾经和我说过这样一件事，他所在的公司有一个产品经理，曾经因为和某程序员就一个Bug的修改而发生口角，最终上演了一出全武行，结果事情发生后的第二天两人就双双离职。

从这个事情中可以看出，作为一名产品经理，个人修养好坏其实是你成熟与否的一个主要表现。在产品管理工作中，个人修养的最核心表现就是“尊重团队中的每个人，相信团队中的每个人”。

而产品经理如果缺乏这种修养，最终损害的除了公司之外，还有就是自己的职业道路。

其实这是一件小事，对于有着良好个人修养的产品经理来说，是能够很好地处理的。但在许多企业，尤其是IT企业中，年轻人占据多数，而年轻人在某件事情上往往易于冲动，因此，加强个人修养的锻炼，对年轻的产品经理来说是需要长期重视和有意识地磨炼的。

创新意识

现在，国内几乎所有的企业都在提创新，但始终没有提出一个创新的主导群体。我们认为，在企业中，创新的主导群体非产品经理莫属。

这是由两个方面的因素决定的。

■ 市场环境：必须承认，现在的市场竞争本质就是企业综合资源的竞争，但企业资源的有限决定了企业必须“把有限的资源投入到最有价值的产品上去”才可以。

而毫无疑问，能够比竞争对手更好地满足市场需求的产品才是最有价值的产品。

因此，从这个角度来说，对于大部分企业，创新的本质就是对市场需求的创造性满足。而这个满足需求的过程，肯定是由产品经理来主导。

■ 职责定位：产品经理在企业中的根本工作，其中很重要的一点就是“让企业持久的盈利”，而其他工作职责都是为此服务而已。持久盈利靠的就是对市场需求的最大化满足。在企业中，唯有产品经理最了解市场需求。

因此，企业的发展靠创新，创新的主导靠产品经理，创新的方向不可脱离市场的需求，市场的发展必须依赖产品的创新，这是相辅相成的。一个产品经理的个人创新能力，首先不是表现在创新能力，而是创新意识上，这个意识的强弱决定了企业产品发展后劲的强弱。

对于产品经理而言，创新首先是一种意识，然后才是能力。伟大的想法来源于现实，现实则来源于市场现实和潜在的问题，而市场现实和潜在的问题则来源于整个业务流程当中出现的不足，正因为有了手动刮胡刀，才有了刮胡泡的诞生。对于产品经理来说，我们要发现的不足不仅包括产品介质本身的，更应该包括整个业务流程当中出现的，这才是我们创新意识的来源。

发现现实的不足并改进它，这是产品经理的职责；发现潜在的不足并知道未来如何改进它，这是产品经理的价值。

沟通能力

我们曾经做过一个调查，在问到“你认为具备怎样的条件才是优秀的产品经理”时，几乎所有人都把“沟通能力”放在了第一位，可见，沟通能力的强弱已经成为企业和个人衡量产品经理是否优秀的第一能力，如图2所示。之所以这样，依然是由产品经理的的定位决定的。

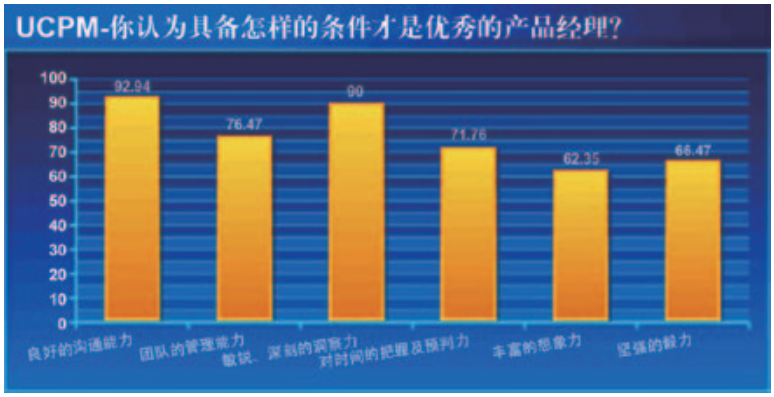


图2 “优秀的产品经理”调查

产品经理“职责不对等”的工作现状，决定了要想很好地开展工作，什么都可以靠，唯有权力靠不上。但在一个部门众多、体系严谨的组织中，缺乏了相应权力的支持，开展工作的难度可想而知。因此，从这个角度来说，产品经理本质就是在靠“嘴”工作，和上级谈、和平级谈、和下级谈、和内部同事谈、和外部客户谈。总之，日常工作的一半以上时间都要花在沟通上。就像一个“传教士”，只有靠自己的嘴去说服他人来支持你的想法，成为想法的信徒。

因此，沟通能力的欠缺，就代表着产品经理连最基本的工作都无法开展。怪不得，有人笑称“产品经理就是谈判专家”……

除语言沟通外，还有一种沟通形式就是书面。其实准确来说，书面沟通只是语言沟通的文字化呈现而已。因为我们习惯于把沟通仅限于语言层面，而对于书面沟通则不太重视，因此，在这里我就只谈一下书面沟通的事情。

书面沟通，顾名思义，就是通过书面文件的形式来传递相关的信息，尤其对于产品经理来说，这点是非常重要的。因为一方面你需要通过相关的文档来记录高层的想法并细化出来，另一方面还需要通过这些文档来指导团队的工作，也就是说，产品经理要撰写的文档既是想法的记录，又是实践的指导。

这就要求产品经理在通过书面进行沟通时，务必要做到“准确、全面、到位”，说到

底，就是达到三个标准就可以了：知道相关文档写什么、知道相关文档怎么写、知道相关文档写到什么程度。

要做到这一点，首先要求企业必须有一个完整的文档管理体系，这是首先书面沟通的前提，然后才是产品经理个人书写相关文档的能力的要求。关于这一点，企业和个人都可以认真评估一下。

抗压能力

工作压力是每一个职位都会存在的，而一个优秀的产品经理所承受的工作压力不会比老总小多少，也正是恰恰因为不是老总，但压力不亚于老总，成就了产品经理的优秀。

这里不去说如何来衡量和缓解压力，只说一下如果没有足够的抗压能力，会给产品管理工作带来什么样的影响。

压力过大，在工作中最明显的表现就是“易怒”，而这种易怒的潜在原因是因为个人对外界事物的敏感性开始增强造成的。

因此，如果压力过大，除了给自身带来许多问题外，一旦把这种情绪带到产品团队中，将会对整个团队的工作情绪造成极大的伤害。并且，如果抗压能力不强，很容易让你的思想混乱，作出一些错误的判断。

由此可见，工作压力承受能力的大小不但会影响到个人的工作，还会影响到整个团队的工作。因此，作为产品经理，一定要具备比其他人强很多的承压能力，否则，在别人还没把你压跨之前，自己就把自己压垮了。

现在很多产品经理都是年轻人，年轻人容易头脑发热，容易冲动，遇到事情不能冷静全面地思考，其实这是做产品经理的大忌。在国外，很多产品经理都是职场和商场中的老油条，但国内毕竟有自身的特点，也必须承认这种现实。

但在这里提一点的就是，年轻人太脆，很容易被压断，要想不被压断，把自己磨炼得柔韧一些是必需的。至于怎样做，其实也没有什么太好的办法，除了时间来历练外，就是作为产品经理，必须要有一个“看得开、拿得起、放得下”的心态，事事都较真注定是做不好产品经理的。

自管能力

自管能力，就是指自我管理能力。

产品经理的事情太多了，缺乏自我管理能力很容易让自己做许多事倍功半的工作。

不可否认，产品经理的工作过于繁琐和杂乱，甚至会有许多本来和自身工作无关的事务来干扰自己正常的工作。这个时候，自管能力的高低就会直接影响到个人工作的成效，或者说，就是产品经理在日常工作中能否承受得住各种各样的诱惑而不会偏离方向。

朋友在一家互联网公司做产品经理，前段时间抱怨说他简直筋疲力尽了。而实际情况听起来都有些好笑：他兼职给客服部做客服。

怎么回事呢？很简单，随着公司用户量的增加，客服部扩招了很多新人，其中很多人对业务都不是太熟悉。当然，在前期，产品部进行必要的支持是没有问题的，但也得有个限度。客服部可不这么想，估计是用产品经理用顺手了，但凡和产品有点关系的事情，就一个电话转接到产品经理那儿。由于朋友负责的是基础产品，因此，被“点”的次数比较多。这还不算，有时还得面临“出台”的情况，直接到客服部去做支持。而朋友脸皮还薄，人家让“出台”就得去，客服部也感觉这是情理之中的。就这样持续了大概一个月，实在受不了了，每天的时间就这样被一块块地切割完了，没法指望做更重要的事情。

从这个案例中可以看出，自管能力其实对自己是一个考验，大道理不说，至少有一点是需要考验的，就是作为一个产品经理，你是否有说“不”的勇气和能力。

之所以涉及到勇气，是因为有很多产品经理和我这个朋友一样，脸皮都很薄，几乎不会说“不”，尤其是面对客服部这样女性比较多的部门，根本拉不下脸来。

之所以涉及到能力，是因为你总得明白或者还需要告诉对方，你为什么要说“不”。无理由的拒绝只会让你的地位一落千丈，而有理有据的拒绝才是产品经理保持地位和让人继续支持的前提。而做到有理有据，绝对是一种能力。这就是自我管理能力，简单地说，就是要求产品经理保证每天知道“说什么、做什么、怎么做”。

产品经理的“软实力”：管理知识

产品管理发源于市场营销领域，随着市场环境的变化，开始与越来越多的学科产生交互，并逐步形成一门新的管理学科。在这个逐渐形成的过程中，融入了许多其他学科中有价值的思想。作为产品经理来说，在管理技能上，除了本身的产品管理技能外，还要学习交互学科的管理知识才可以。

原因在于，现在的产品经理已经被赋予了新的内容，不再是市场的附属，越来越多的企业开始把产品经理提升到一个“一人之下，万人之上”的地位，这种期望就要求产品经理必须具备全面的管理知识才可以，否则，将永远不能成为“真正的CEO”。

我们认为，除了必备的产品管理技能外，产品经理还需要具备的管理知识应该包括四个方面。

- 战略管理知识
- 项目管理知识
- 时间管理知识
- 团队管理知识

这四项管理知识是和每个产品经理都息息相关的。或许这四项管理知识不会直接影响到具体的工作，但它们会通过影响产品管理技能而影响具体的工作。因此，这四种知识是否具备将会间接影响到本职工作。

于是，我们把管理知识称为产品经理的“软实力”。

战略管理

战略并不是很虚或者可有可无的，国外曾经对产品经理做过一个调查，调查问“你认为公司制订的战略对现实工作有什么帮助”，大部分的被调查者认为“虽然战略对现实的工作并不能提供直接的帮助，却可以让我们更为有目标和有方向地进行工作，能够让我们更容易知道未来要做什么”。

关于战略，大道理就不讲了，从这个调查中可以看出，学一些战略管理方面的知识，应该至少有三点作用。

- 目标：通过制订一个战略，可以为业务的发展确定一个清晰的目标。
- 方向：通过制订一个战略，可以让业务的执行人知道应该开展什么样的工作。
- 节奏：通过制订一个战略，可以对业务发展的进度进行相应的管理和控制。

我们可以试着回答一下这三个问题。

- 你是否对产品的长期发展有足够的思考？
- 你是否知道5年内产品的走向？
- 你是否明确5年内的问题和机会都是什么？

如果你不能清晰回答这三个问题，那只能说明你还没有用战略的眼光去看待你的产品。

许多产品经理可以知道1年内要做什么，但如果回答不了以上3个问题，那问题就比较明显了。

有朋友肯定会说：“那不是我要考虑的。”那么，什么才是你要考虑的呢？

需求分析？产品定义？还是销售支持？没错，这些都是你该考虑的事情。但即使把这些都做得非常好了，在这些方面的能力也都非常强了，你充其量还只是一个“将才”，而不是“帅才”。

说到底，战略制订其实就是“基于现在对未来的一种预测和评估，并提出相应方案”。一个企业、一个产品如果缺乏这个工作的支持，产品经理缺乏这种知识的协助，那么将无

法对未来做有效的预测，或者出现错误的预测，那么，输掉的不仅仅是一个产品，而是一个市场。

其实，战略做得好不好，取决于两点：你能看多远、你能看多广。

项目管理

这里不去探讨产品管理和项目管理的联系和区别，重点说一下为什么产品经理要学习项目管理的知识。

从过程上看，产品其实就是由一个项目阶段加一个项目阶段组成的，就是说，产品经理负责的是过程，而项目经理负责的是阶段。

由此可以看出，作为产品经理，在产品管理工作中，如果用“天”作为工作计时单位的话，每天的工作就是泡在某个阶段的项目中。

做市场调查，对于市场部门来说是一个项目；需求定义，对于产品部门来说是一个项目；产品研发对于研发部门来说，同样也是一个项目。

就是这样众多大大小小的项目构成了一个产品项目周期，而多个产品项目周期则构成了产品的过程。

因此，从这个角度来说，产品管理的工作也可以看成是在做着一个“没有明确时间界定的项目”，而项目管理中需要的各种知识和技能同样适用于此，主要包括：

- 制订项目计划
- 评估项目风险
- 争取项目资源
- 管理项目团队
- 协调项目问题
- 控制项目进度
- ……

因此，产品经理熟悉掌握一些项目管理的知识和技能，对现实的工作有着非常积极的作用。

当然，产品经理的项目管理毕竟和项目经理的项目管理并不完全一致，我们只是采用项目管理的思想来让产品管理的工作更有序、可控，也没有必要花太多的时间和精力去做项目管理的具体工作。当然，如果公司要求你必须扮演产品经理和项目经理的双重角色，那就另当别论了。

总之，作为产品经理，只需要借鉴项目管理的思想，把一个项目的输入、处理和输出这三个关键阶段把握好就可以了。

时间管理

可以这么说，产品经理除了做必要的管理工作外，几乎都是被大量的和产品有关的日常事务所充斥，但一个人的时间和精力毕竟有限，如何在有限的时间内尽量多地完成这些工作，就涉及到了时间管理的知识和技能。

在时间管理中，有两个最重要的概念：效率和效果。

- **效率**：对于产品经理而言，就是指我们在现实的工作中，花费时间和有效时间的一种比例。花费时间是总时间量，比如我们一天工作了8个小时，那么，在这8个小时中有多少时间是在按照我们的计划来开展的？这种按照计划执行的时间就是有效时间，这种对有效时间的管理就是我们的效率。
- **效果**：但效率有高有低，这就涉及到了第二个概念——效果，也就是指在有效时间内我们对相关事务工作顺序的排序，常见的就是按照事情的“重要”和“紧急”来形成一个矩阵，然后根据矩阵中的事务排序来开展工作，这就是我们的产出效果。

可以用一句话总结一下：**效率就是在有效时间内知道做哪些事情（即上面提到的计划内的事情）；效果就是在有效时间内知道如何做好这些事情（即上面提到的如何区分计划内事情的“重要性”和“紧急性”）。**

也可以这么说，时间管理的目标就是在“效率”和“效果”上达到最适合的平衡。

但现实的情况是，现在几乎所有的产品经理都在扮演“消防员”的角色，到处救“产品的火”，一天下来，累得要死，效果却不尽如人意，还往往把重要的事放在了一边，被高层认为是“既没有效率，又没有效果”。

要改变这种情况，唯一的方法就是把“救

火”变成“预防”。其实，在日常的救火工作中，你会发现，有许多火是同一原因引起的，这时你就要想一些办法了。最常见的方法是“压缩法”，即把所有相同原因引起的问题整理出来，压缩成一个问题，然后提交给相关部门。这样，一旦再遇到这些问题时，它们会通过“解压缩”的方法来自我救助，而不是首先想到的是产品经理。

关于时间管理，一直是个流行的话题，这方面的图书和资料也很多，如果大家有兴趣，可以好好学习一下。

团队管理

众所周知，产品经理绩效的产生依赖于业务团队的工作，因此，能否对业务团队进行有效的管理，将直接影响到产品经理工作的成绩。

团队好不好管理呢？说实话，对于国内大部分的产品经理来说，普遍认为是非常不好管理的。

最典型的莫过于就是你如何让相关业务团队能够遵从于产品目标而不仅仅是团队成员个人的绩效目标。

记得2005年时，我在一家互联网公司做产品经理，当时负责公司的一款基础产品。公司为了改善这款产品，投入了很大的力量，成立了一支40多人的研发团队，这个团队由一个项目经理负责，而产品经理负责的则是产品的设计和进度的跟进与资源的协调。

按理说，业务团队自然是以业务目标（例如产品是否实现了规格所定义的指标、是否能够如期上线等）为主的，而业务目标自然也是由产品经理来控制的。

但问题出现了，项目经理要为进度和质量负责，在遇到一个技术问题时，他通常会基于这两点来思考如何改进。那么，就很容易出现这样一种情况——改进的思路和方法可能会影响产品规格指标的实现。

作为开发人员，这个时候应该听谁的呢？

同样是按理说，应该是听产品经理的。但事实是他们几乎不会去听产品经理的，因为他们的月度绩效是由项目经理来考评的，如果不听项目经理的，那么月度绩效则会受到一定的影响，说得通俗点儿，就是每个月领到的工资会有变化。

因此，那个时候我们就显得很纠结，定义好的产品规格却不能圆满实现，出现问题则几乎无法对开发团队进行有效的干预，本来并不是复杂的事情，往往要通过各自的老大沟通才能解决。

在这样的情况下，效率和效果又从何谈起。

但作为产品经理，至少在短期内你必须面临这样的团队管理问题，怎样能做好呢？抛开公司体系可能出现的影响，仅从个人来说，磨炼一些团队管理的技巧是必不可少的。

例如，如何组建合适的团队、团队成员遇到问题时作为个人应该如何不通过权力或者上级压力来解决，都是我们需要着力去加强的技能。



产品经理的“竞争力”：核心能力

产品经理的工作，就是运用各种管理知识和技能来为企业的产品发展甚至于企业发展服务，在这些知识和技能中，最为核心的就是自身的产品管理方面的知识和技能。在产品管理刚兴起时，市场是处于“以销售为中心”的阶段，产品经理的主要立足点是“产品生命周期”的管理，主要任务就是把企业生产的产品销售出去，而无需太多考虑用户的需求。但随着市场环境的不断发展，在进入到“以市场/消费者为中心”的阶段后，用户需求成为主导企业生产的第一要素，因此，只把眼光放到产品生命周期管理上，已经不能满足这个阶段的需要，越来越多的管理内容被加到产品经理的身上。

我们认为一个产品经理，在产品管理核心知识上，应该具有以下五个方面。

- 产品规范管理
- 产品需求管理
- 产品项目管理
- 产品周期管理
- 产品品牌管理

产品规范管理

首先解释两个概念，产品标准和产品规范。标准和规范的最大区别就是，标准是由官方制订，并且具有一方强制性的约束条件；而规范则是由民间制订，不具有约束性的自我遵守的条件。

具体到产品标准和产品规范上，可以做如下理解。

产品标准通常会在产品信息上有明显说明，例如国家标准（GB）、卫生标准、质量安全（QS）等，这些都是要强制执行，并且由专门的国家机关来管理，企业只是执行方而已，并且这些标准通常不被用户所熟知。

产品规范通常采用直观的形式加以说明，例如瓶装水，容量就完全由企业来决定，可以有600ml、550ml、595ml等多种容量规格，也可以在瓶口直径上有自己的规格，例如有2.5cm、2cm等规格，这些规范对于企业来说没有强制性的要求，完全由企业自行管理。通常规范的制订来源于大多数用户的需求，同时，企业也会主动来增加一些产品规范来进行用户引导，并且这种规范通常会被用户所熟知。如表1所示。

表1　产品标准与产品规范的对比

	产品标准	产品规范
制订方	国家机关/组织	企业；用户
执行方	企业	企业
管理方	国家机关	企业
是否强制执行	是	否
用户是否熟知	否	是
产品体现	不明显	明显

由此可以看出，产品规范是任何产品都不可或缺的一种信息。同时，随着产品多样化的不断发展，产品规范会越来越丰富，也逐渐成为企业的一种营销手段。在此情况下，企业就必须有专人来对这些产品规范进行管理，这个责任自然而然就落到了产品经理身上。

产品规范管理通常包括以下几个方面。

- 产品规格：主要包括容量、体积、尺寸等，例如瓶装水的1.5L容量。
- 产品外观：主要包括规则外观和非规则外观，例如NOKIA的枫叶外观手机。
- 产品包装：主要包括色调、图案、文字等，例如可口可乐的红色调丝带包装。
- 产品介质：主要包括有形介质和无形介质，例如软件的光盘传播介质、网络传播介质。

■ 产品提示：主要包括产品附件、用户手册、售后服务手册等，例如冰箱的使用说明。

■ 产品文化：主要包括产品所包含的非物质性含义，例如可口可乐公司为纪念可乐推出100周年而推出的纪念性可乐。产品文化不能独立存在，必须依赖于某个载体，例如包装。

说了这么多，可能大家还是有些不理解，下面举一个大家都熟悉的例子。

我们在进行产品商品化包装时，肯定会涉及到一项工作，就是制作产品的说明书和帮助文件，这就涉及到产品规范管理中的产品提示。

我记得以前做这类文件时，有一本厚厚的撰写规范说明作为指导，主标题应该用几号字，副标题应该用几号字，正文应该用几号字，图片应该用什么格式，在印刷版和CHM版中的图片格式都是什么，说明书的结构应该怎样，图片应该如何截取，在图片上应该如何添加文字等。

这样做的好处就是把很有可能个人随意化的工作变成了规范的流程化的工作，这个工作看起来好像不太起眼，却是公司形象的一种体现。作为产品经理，难道不需要关注和规范管理起来吗?

产品需求管理

需求已经谈得很多了，却很少有人从产品管理的角度去谈到底什么是产品经理应该关注的需求。

很多对于需求的理解都是基于传统的产品本身甚至是开发层面的，这些需求是不是产品经理应该关注的呢？没错，确实是应该产品经理关注的，但如果只去关注这些需求，只能让产品经理对需求的管理产生片面的理解。

对于产品经理而言，需求需要从战略、规划和战术三大活动来关注，如图1所示。



图1 产品经理的三大活动

下面举例说明这三类需求的关系。

我曾经在某企业负责过一个基于移动终端的产品，那么，对于企业来说，肯定是有一些期望在里面的。这个期望对于企业来说，可以称之为是企业的一个目标，而对于我来说，则是一种需求，原因很简单，因为企业的期望所包含的是一种诉求，企业希望你能够帮助它实现这个期望。

这就是需求的第一级别：**商业需求**。

针对这个需求，你要明白企业想通过这个产品实现什么，是为了更好地实现企业的战略目标甚至是愿景，还是希望通过这个产品增加营收，或是延伸现有的产品线，开拓一个新的市场，甚至就是纯粹的资本运作，为获得融资以及可能的IPO服务。

清晰了这个商业层面的需求，接下来你就应该知道如何来规划这个产品了，这就是**业务需求**。业务需求要实现的就是如何来规划这个产品从而更好地有助于实现企业的商业上的期望。

如果企业的商业需求是为了开拓一个新的市场，那么，这个需求针对的业务需求就是要求你考虑和规划能够实现这个目标的产品方向和战略。

你或许需要考虑如果达到既定的市场目标，可能需要5年时间。那么在这5年中，你需要对产品进行怎样的规划，公司需要的投入都有哪些以及数量，每个阶段的产品目标都是什么，目标之间应该存在什么关系等。

这些考虑到位了，接下来的就是按照既定的计划来设计要实现相应目标的产品。比如第一年的目标是让潜在用户对新产品有一个大致的认识，在市场上树立一个品牌，公司的相关投入也只是尝试性的，那么，你在具体的工作中就需要围绕这个目标来开展，比如对于产品的功能性需求的满足就会缩减，可能只需要实现基础功能（按照UCPM对功能的分类，可以分为基础功能、延伸功能、增值功能）就可以了。

因为公司的投入是尝试性的，并且其目标主要是为了树立品牌和建立潜在用户的认知，因此，更多的投入不会在产品本身上，而是在推广和营销以及相关事务的建设上，例如新渠

道的构建等。

这就是**项目需求**，因为要实现业务需求，肯定是需要一项项的工作来完成的，而这一项项的工作对于产品经理来说，则是一个个具体的项目。

其实对于产品经理来说，关注的需求到这里就可以了。当然，再往下，还可以细分出更多的需求：基于项目需求还会有产品具体版本的需求、基于版本需求还会有开发层面的需求，这些在产品管理中，其实并不需要作为重点。

按照产品管理规范的流程，产品经理在完成MRD和PRD后，需求的工作其实是可以告一段落了。

再强调一点，我们通常所说的“用户需求”、“市场需求”等，其实都是包含于以上三种需求中。

原因很简单，一个产品其实是承载了企业和用户两方面的诉求在里面的，我们要做的就是能够把这两类诉求统一起来，用一个产品来满足双方的期望和目标。

而商业需求、业务需求、项目需求则可以把这两类需求同时让产品经理关注起来。

这其实是产品管理的目的所要求的，产品管理的目的就是“保证企业的竞争优势和客户的长期满意度”。

明白了这点，其实就明白了如何做好需求管理。

关于需求管理的工作，这里不再详细说明，简单来说，大致可以分为六步，如图2所示。



图2 需求管理工作的步骤

产品项目管理

在《产品经理的“软实力”：管理知识》一文中，已经讲了产品经理应该掌握一定的项目管理知识。那么，本文中产品项目管理又指什么呢？是不是和项目管理知识是一回事呢？

自然不是了，掌握项目管理知识是完全从项目管理的角度来看的，也就是说，我们必须学习项目管理知识，而产品项目管理则是完全从产品管理的角度来看的——这个知识指的是要用从项目管理中学习到的知识来指导产品管理中的具体工作。

简单来说，前文中的项目管理知识是理论基础，而这里的产品项目管理则是工作实践。

因此，表2总结了产品管理中，涉及到的管理工作都有哪些。

表2 产品管理涉及的管理工作

战略活动	找到市场问题，评估能够解决这些问题的所在组织的商业资源。知道我们要做什么。
预测问题	（捕捉）市场问题
发现机会	（分析）商业需求
	（分析）竞争格局
	（确定）独特能力
	（评估）技术实力
创新产品	（确定）目标市场
	（管理）产品组合
	（管理）产品创意
	（制订）商业计划

规划活动	提供完整的市场问题解决方案和行动计划并使其纳入组织目标和计划中。知道我们如何去做。
产品路线	（制订）价格策略
	（分析）产品收益
	（明确）分销策略
	（制订）产品路线
产品策略	（明确）产品定位
	（构建）业务流程
	（塑造）用户原型
	（制订）年度策略
产品计划	（制订）营销计划
	（制订）状态指标
	（制订）年度计划

战术活动	
概念化	（完成）概念筛选
图纸化	（制订）产品规格
技术化	（完成）产品验收
商品化	（制作）销售工具
	（开展）销售培训
	（完成）渠道准备
	（制订）推广策略
	（明确）销售流程
	（制订）服务方案
	（制订）发布计划
市场化	（完成）产品发布
	（提供）渠道支持
	（提供）事务支持
	（提供）销售支持

产品周期管理

这里之所以没有说产品生命周期管理，是因为按照通常的理解，产品生命周期是指“一种产品在市场上出现、发展到最后被淘汰的过程，它是产品的一种更新换代的经济现象”。

也就是说，一个产品的生命周期是以这个产品的上市为开始，退市为结束的。但对于产品经理来说，现实的情况却不仅仅于此，要做的工作不但包括上市到退市这个过程，而且还包括前期的很多工作。也就是说，产品管理过程不仅限于产品生命周期管理，产品生命周期管理只不过是产品经理的一个阶段工作而已。

这样说，似乎还是有些过于学术。说得简单点，产品周期管理包括做产品和卖产品两个阶段。

而做产品阶段包含了产品项目管理中的大部分工作（除市场化以外的工作），卖产品阶段则是战术活动中市场化的工作。那为什么我们提产品生命周期管理比较多呢?

这是因为产品生命周期管理是产品周期管理中实现市场价值交换的阶段，现实地说，这个阶段对企业来说是最具吸引力和诱惑的，毕竟是要实现收入的过程。因此，这就造成了企业非常重视这个阶段的工作。

企业现在非常重视的营销、推广、各种促销、各种策划等，都是处于产品生命周期管理这个阶段中。

除了这个原因，还有就是国内的企业一直以来还是以“销售为导向”进行生产，对于前期的战略和规划并不是很重视，认为只要营销做得好，没有产品卖不好。基于这样一个思路，肯定不会有一个整体和长远的考虑的。

但市场是要发展的，以销售为导向的生产必然要被以市场为导向的生产所取代，这是一种必然的趋势。因此，企业肯定会逐步重视起前期的工作来，而我们这些产品经理需要做的就是做好这个准备，储备好必要的知识和技能，要最终做到召之即来，来之能战，战之能胜。

这里只简单说一下，大家只需要知道“产品周期>产品生命周期”就可以了。做产品经理的，不但要懂如何卖好产品，还要懂如何才能做好产品。我们常说“适销对路”，其实应该是“对路适销”才对。因为“对路”是前提，也就是把产品做好了，而“适销”是结果，也就是把产品卖好了。

这样理解就更简单了，产品经理无非就是把这两个工作做好，但想做好，那是需要下大力气的。

产品品牌管理

自从2008年的金融危机以及国家提出产业升级后，我们会发现各类媒体以及很多原来是做代工的企业都在强调一个理念，就是要走创立品牌的道路。

有这个理念是好的，但从想法到实践再到达到目标，这中间要经历的各种风险和成本不知这些企业是否已经做好了必要的心理准备。

从企业的角度看，品牌是一种以产品介质为基础的符号；从用户的角度看，品牌则是用户对产品的一种情感认同，从一个冷冰冰的符号变为用户心理层面的一种价值，大家可以想想有多么难吧。

我们曾和广州一家做小家电的朋友聊天，这家企业一直是做代工的，并没有自己的产品品牌。外贸订单减少后，企业也开始考虑内销，当然，产品是现成的，在他们看来，可能需要的就是创建一个产品的品牌就可以了。

但品牌的创建就是这么简单吗?

自然不是，创建品牌几乎和规划一个产品一样困难。

就拿最简单的设计品牌名称来说，一个名称可不是坐到办公室里就能琢磨出来的，同样需要做很多分析才行。

■ 首先你要确定你的目标用户，因为目标用户的特征会影响他们对你品牌的认知和接受度。例如，如果你的产品的目标用户是生活在农村的中老年人，那么，你在设计品牌名称时，就肯定不能过于复杂和洋化，最好是一个符合农村文化特征的牌子，还得简单、好懂、易传播。

■ 其次，在确定名称后，还要考虑配套的Logo、Slogan、传播途径、传播形式，以及如何与竞争者的品牌有所差异、在品牌推出后，如何评估品牌的效果等。

这里面涉及到哪些工作了呢?

至少包括市场细分、目标客户确定、竞争格局分析（品牌层面的）、产品品牌创意、差

异性分析、营销策略制订（品牌层面的）、推广方案制订、广告设计、传播渠道设计等，至于后期的品牌维护则又是一个大头的工作。

由此可以看出，做品牌管理不是简单的设计个品牌，制作个广告，媒体上一放就可以了，同样需要大量细致的工作去支撑。

当然，产品经理不可能，也没有必要把上面的工作一一做到，很多工作还是需要团队去配合的。但产品经理在品牌管理中至少应该向企业和团队说明以下情况。

- 我们的品牌要传播给谁？
- 我们的品牌如何才能与竞争者有所差异？
- 我们品牌要传递什么信息（质量？服务？价格？心理满足？自我实现？）给目标用户？
- 我们的品牌如何才能传递到位？
- 如何评估品牌的传播效果？
- 如何维护我们的品牌不受到侵害？

这还仅仅是针对一个新品牌诞生要做的工作，如果是对已有品牌进行管理，那工作也是不少的。例如，现在推出了一个新产品，你就要考虑现有的产品品牌是否能够支撑这个新产品的发展，如果能够，那么应该如何来延伸现有的品牌；如果不能，是否需要重新设计一个新品牌。此外，如果企业已经有了多级的品牌，还要考虑延伸或者新的品牌和不同级别品牌之间应该是一种什么关系才会是促进与和谐的关系，而不是影响的关系。

上面这都是产品经理的品牌管理工作应该要做的。

最后，可能有朋友会问了，说了这么多，那你能否告诉我产品经理为什么要去做品牌管理工作呢？

其实就一点：我们做产品，就是对企业的资源进行管理，而品牌作为产品不可缺失的一种企业资源（在国外，品牌是作为一种战略资源存在的），自然要成为我们关注和管理好的一个重要方面。

结束语

其实产品经理需要了解的知识是很多的，而这种知识则构成了一个产品经理成长的基础，缺失了某一方面的知识，都有可能影响到自己在这个职位上的发展。

对于我们这些从业者来说，可以让我们对自己进行一个评估，找到自己的不足，并且在需要弥补的方面加以努力，而对于那些有兴趣成为产品经理的人，则可以把这些方面作为自我评估的指标，来判断自己是否适合做一名产品经理。

对于企业这些用人单位来说，则可以更加清楚应该如何帮助产品经理设计合理的职业发展路线并提供必要的支持和知识，同时也可以作为考评产品经理的一个指标。

当然，产品经理所需的知识也是随着市场和企业的发展而不断变化的，我们的知识不能停留在一个水平上就停滞不动。只有不断的发展、完善和提高，我们才可能越来越接近“优秀”。

关于UCPM

UCPM（中国产品经理联盟）是一家专为产品管理人群和企业服务的专业组织。服务的人群包括与产品管理相关的产品经理（专员）、产品总监（产品部经理）、企业高层和对国内产品管理体系关注的人士，所服务的人群遍及国内所有行业和地区，到目前为止，UCPM注册会员已经超过70000人，并得到了国内众多产品管理人员和企业的认可，已经成为在产品管理领域最值得关注的、大型的、专业的职业人群专业机构。

UCPM中国产品经理联盟的定位是“交流、学习、发展、生活”，希望为产品管理人群提供一个产品管理交流和服务的平台。UCPM在两年的发展时间内，已经逐步和越来越多的企业、培训机构、学术团体和媒体建立了良好的关系，旨在通过联盟线上和线下的各种活动的开展，为产品管理相关人士提供获取知识、个人信息交流的平台，积极促进和推动产品管理在中国的发展与进步。

互联网产品经理历练之路

文 / 刘舒 张婷婷

正如一千个人眼中有一千个哈姆雷特一样，如果现在要给互联网产品经理下一个定义，未免过于武断。也许我们还无法想象产品经理这一角色可以使产品成为什么样子，但我们可以来想象，如果没有产品经理，产品会是何种样子。

下面以一个场景为例进行说明。

某公司老板鉴于国内外智能手机市场的火热以及移动应用商店的重要性，决意打造一个移动应用生产、流通及交易的电子商务平台，并高瞻远瞩地定位该应用商店为国内最安全的、精品的以及让开发者获益的平台，同时决定，该应用商店将在6个月后发布上线。

此规划一出可难为到了所有的项目相关人员。

- UI设计人员：产品方向太大、太泛，没有功能定义，不知设计什么，更不知该从何设计。
- 开发人员：没有功能说明和UI设计，不知道该如何将愿景转化为产品，无从编码。
- 测试人员：不明确产品的完整功能以及符合要求的产品标准是什么，缺少测试依据，无法进行测试。
- 运维人员：没有可部署上线的东西，无事可做。
- 运营人员：无法预知产品的庐山真面目，无法提前开始前期宣传以及其他支持推动工作。
- 设备厂商：无法预知产品的开发时间，无法预装和保障产品的推广。

此外，所有项目相关团队的独立运作，导致了彼此信息的断裂和不对称，无法保证项目在既定的时间内开发完成，更无法保证开发出的东西与原始需求相一致。于是，产品经理横空出世，他规划和定义产品形态，推动产品的研发以及收集分析产品反馈，周期性迭代地对产品进行优化更新，其关系如图1所示。

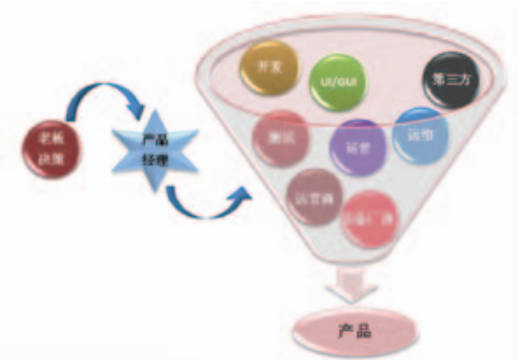


图1 互联网产品经理的角色图

别怕，产品经理是可以养成的

产品经理应该是一个产品的设计者、执行者以及润滑剂。产品经理如此重要，但没人生来就是产品经理。卓越的产品经理都是通过系统的训练以及实际工作中的不断积累历练出来的。

产品经理的成功要素，归纳起来集中于四个方面的能力，即产品的规划、定义、执行和反馈，其实这四个方面也是一个互联网产品的全生命周期。其关系如图2所示。



图2 产品经理的能力模型

规划

所谓规划，理解起来很简单，即是解决产品的“what”（做什么）和“how”（如何做）的问题。但要真正做出可执行、可发展的规划，则对产品经理有着较高的要求。

- **行业视野及洞察力**

任何一个互联网服务产品，不会是凭空创造出来的。产品一定是根据行业市场的某种需求，通过互联网的手段，巧妙地构建和运营出来一种服务模式，以此来满足市场和用户的需求。因此，任何互联网服务，第一步是要想清楚做什么（what），而一个优秀的产品经理，应该具有宽广的行业视野以及敏锐的洞察力，才能做好产品定位。

什么是行业视野？简单地说，即是对行业市场及目标用户的准确分析和把握。虽然行业无好坏之分，但定位决定产品成败，不同的细分市场定位、产品定位、渠道定位以及目标群体定位，就足以决定和预见一个产品的发展趋势。而对于产品经理来说，其行业视野的大与小，也就决定了其产品日后的走势。拿淘宝网和团购网站来说，虽然二者皆是电子商务平台，都打着“省钱”的口号，却有着不同的商业模式及渠道模式，而这种不同，也将最终决定它们未来的发展趋势。作为电子商务网站龙头的淘宝网，其“卖家”涵盖了虚拟的网络商店以及实体品牌店两种，结合了C2C与B2C两种经营模式；同时，所提供的服务几乎覆盖了人们生活中衣食住行的方方面面，它的市场定位很广泛，因此有着大淘宝的概念。而对于团购网站，它所面对的市场相对狭小，仅为大淘宝市场中的一个细分子市场，并且，其针对与实体店铺的经营模式，也决定了其合作商及用户数量的局限性。因此，从长远的市场发展以及目标消费群而言，淘宝网较团购网站有着更好的发展前景。

行业视野及洞察力的养成是需要积累的。优秀的产品经理会注重通过各种途径（杂志、新闻、各种行业讨论、分析报告等）了解各行业的发展情况，也可以通过分析公司现有及历史的市场数据、用户行为等特征，了解相关行业及公司的发展状况，这些都是对行业视野及洞察力的提高很有帮助的方法。

■熟悉互联网模式

仅仅具备了独到准确的行业视野及洞察力还是远远不够的。为了说明这个问题，我们还是先回到文章开始提到的构建移动应用商店的故事。确定开发移动应用商店的战略只是产品的一个开始，而这个商店包括哪些东西、流程如何设计、用户如何来获取等问题的解决，才是产品成功的重要保证。

要思考清楚这个问题，我们先将问题分解一下。对于一个应用商店，必须要有买和卖的双方，而商店只是当中的一个载体，那么如何来获取买和卖的两方，对应用商店来说至关重要。在互联网的应用商店经营模式下，“卖方”所指为开发者，如何来吸引开发者呢？也许可以通过为开发者提供开发SDK的方式来为开发者提供一些便利条件；而对于“买方”来说，如何来满足用户的购买呢？网络上的销售要求具备支付平台，网络的推广要求具备广告平台，用户信息的管理要求具备账号系统等。除此之外，如何来增加买卖双方的互动与黏性，可以借助于社区（SNS）的力量来实现。以上便勾勒出了一个应用商店最原始的形态。而其中涉及到的开发者SDK、支付平台、广告平台以及SNS社区等，这些都属于互联网的产品模式。一个优秀的产品经理，应该对互联网产品模式了然于胸，当一个产品的大规划出台后，要能迅速地搭配并巧妙整合各种产品模式，来完成对产品形态的架构。

对于互联网模式的积累，没有任何捷径，唯一的修炼方法就是“多看、多用、多悟”。

多看，是指平日留心各种关于互联网产品的新闻动态及分析报告，了解各种互联网服务模式的特点。

多用，就需要自己作为一名用户，亲自去深入使用各种互联网产品，比如，对于互联网服务四大类：搜索、社交、电子商务以及网游，是必须亲身且深入的使用体验，才能洞悉和了解每类服务其中包含的产品模式，以及灵活应用。

多悟，就是结合自身的使用经验以及研读的资讯信息，将这些产品模式消化，真正变成自己设计产品时的利器。

因此，经过以上两点的分析我们可以看出，一个优秀的产品经理，首先需要具备高瞻远瞩的行业视野及洞察力，并要熟知互联网的商业模式。他不仅要对产品的大方向进行把握，保证产品符合市场需求，更要对产品的具体实现组成进行把握，寓目的于手段之中，勾画整个产品的蓝图。如此，便是产品的规划。

定义

构建一个互联网产品，和建造一套房屋的过程是类似的。有了第一步对房屋的规划后，必须还要有设计图纸，工人才知道所要建造的房子有多少层、每层有什么格局、每户有什么要求，否则，就算再多的理念和愿景摆在他们面前、再多的时间分给他们，他们也无法知道如何开始建造一件房子。同样，一个互联网的产品构建图纸，就称为产品定义。

■信息框架

我们可以把产品的定义看作是在设计一栋房屋的图纸，首先肯定是要先确定它的基本户型，在此基础上，才能设计各个地方的细节。同样，对于产品的定义，我们首先需要定义它的信息框架，确定基本的功能构成。例如，我参与过的一个移动应用商店的基本信息框架如图3所示。在工作中，常用的信息框架的设计工具是MindManager，它可以既简单又清晰地记录各个功能的信息结构。

■功能描述

对于房屋的设计图，在确定了户型结构后，还需要对每个房间的具体布局、各个位置的面积大小等进行设计，这样才能保证工人可以开工建造。对于产品定义也一样，定义了信息框架后，还需要对每一个功能进行详细的设计和功能描述，以便开发与测试人员可以按照设计来进行工作。常用的功能描述方法有用例化和模块化两种：用例化的方法，以角色为出发点，可以清晰地看出各种不同角色可以进行的操作；模块化的方法，以功能模块为出发点，可以保证所有的功能均被涵盖在内而没有遗漏。这两种方法，各有优点，可以根据需要选用其中的一种，也可以两种同时使用。常用的功能描述的工具主要有MS Excel和MS Visio等。

■形成完整产品定义

在进行了信息框架的设计和功能描述后还需要做什么呢？我们可以回想一下，房屋的建造都是需要设计图的，这种图纸不仅可以清晰地表现所有结构和功能，相比单纯的语言叙述也更形象和直观。因此对于产品的定义，也同样需要这样一种设计图，这便是UI设计的作用。对产品的功能描述与这样的UI设计图二者结合起来，才算得

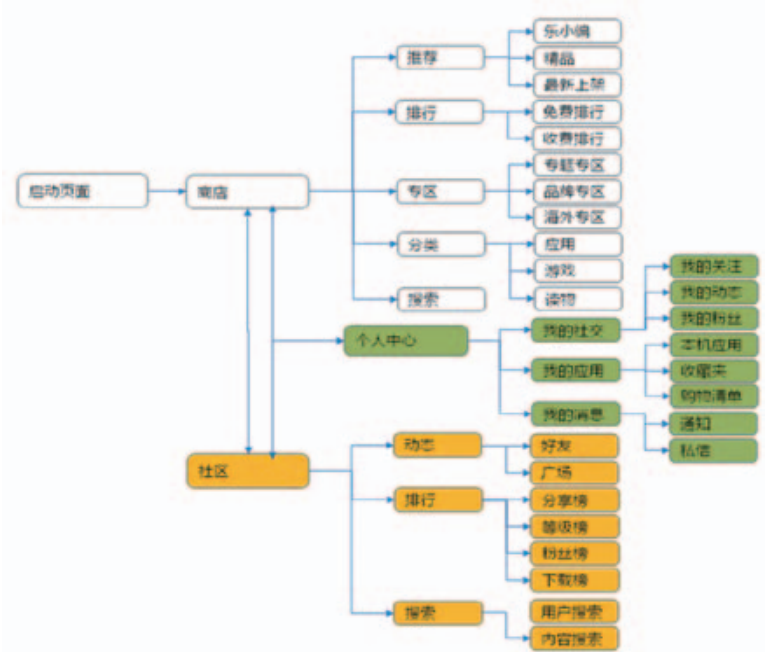


图3 移动应用商店信息框架

上是一份完整的产品定义文档。例如，图4中关于移动应用商店中评论功能的定义，包含了功能描述、UI页面设计及交互设计3个部分组成。

执行

执行也就是项目管理的过程，它要求管理者对项目的开发方式、开发进度以及产品质量进行把控，是产品经理需要具备的核心能力之一。

■研发流程选择

进行项目管理，首先需要准确地选择一种适合于项目特点的开发流程。常用的开发流程有三种，分别为瀑布式开发（Waterfall）、迭代式开发（Spiral）和敏捷式开发（Scrum）。其实这三者并无本质的区别，而仅是在开发反馈周期上有所差异，比如对于Waterfall，开发周期通常为6个月甚至更长，Spiral通常为1~2个月，而Scrum最短，通常仅为1~2周。不同的项目可以根据开发产品的特点而选用不同的开发流程。比如我之前在微软任职时，开发的软件项目，通常采用迭代的开发流程；而在腾讯所进行的一些互联网项目，通常采用Scrum的开发流程，缩短开发周期。

■关键里程碑

仅仅确定了研发流程对保障产品的按时上线是远远不够的，一个资深的产品经理在项目管

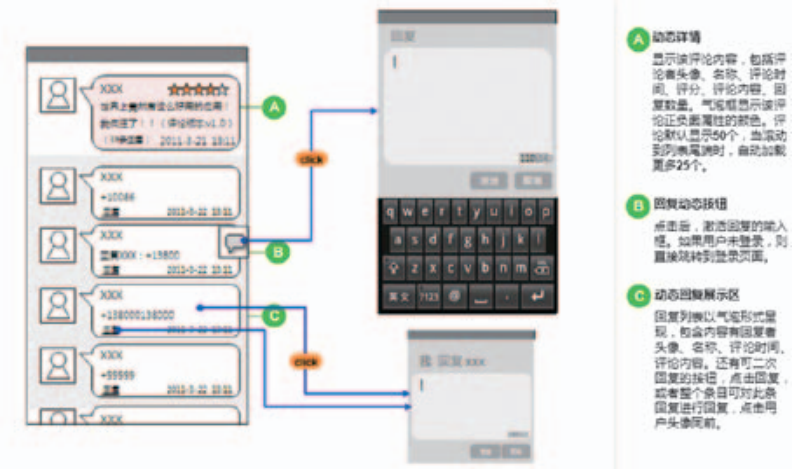


图4 应用商店完整产品定义文档示例

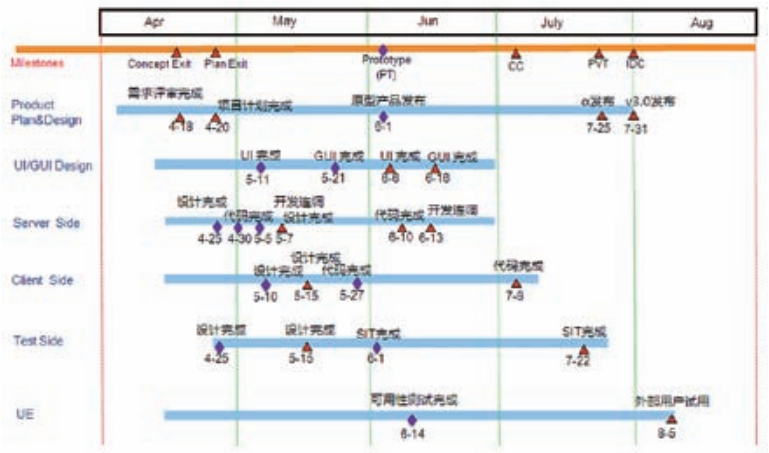


图5 移动应用商店产品研发节奏表

理过程中，最重要的能力是能够定义产品的关键里程碑，并有一套强有力的方式推进研发过程，保障每个里程碑按时达到。通常需要关注的里程碑主要有：设计完成（Desing Complete）、代码完成（Code Complete）、零缺陷边界（ZBB）、内部发布试用（α-release）、外部发布试用（β-release）、正式发布上线（IDC），进而确立产品研发的节奏表。图5展示了某移动应用商店的研发节奏表（Roadmap）。

当然，对于开发流程和里程碑的制定，概念上看起来很简单，但在实际操作中，却需要有较长时间的工作经验做支撑。项目管理的过程是相当琐碎的，很多问题都需要产品经理来做调解和权衡，拿我之前做过的一个产品来说，在产品上线之前的1个月时间里，每天下午的3~5点，都在会议室中，与北京、成都和上海的同事进行缺陷评审（Bug Triage）的工作。现在已经不记得在那些无尽的Bug与电话中度过了多少个晚餐时间了，但这就是产品经理需要进行的工作，很重要的工作。因此作为产品经理，一定要摆正自己的位置，他既是产品相关人员的管理者，也是他们的服务者，需要为大家提供必要的支持以及能解决问题。

■ 工具

在项目管理过程中，善用一些工具，能够极大的促进研发管理效率。根据我自己的经验，推荐一些比较好用的工具。

Redmine：一个免费开源的项目管理工具。管理项目里程碑、工作项等。

Maven+Hudson：用于研发自动构建。

Bugzilla：一个开源的缺管理系统。

Testlink：用于对测试用例及产品测试报告管理系统。

SVN：一个免费的源代码管理系统。

反馈

■ 反馈的意义

产品与项目的不同，主要体现在项目是一个有头有尾的段落，而产品则是一个持续演进的生态系统，它只有开始，没有结束，除非达到它生命周期的末端而退出历史舞台。因此，反馈信息对于产品整个生命周期的推动都有着至关重要的作用。正如腾讯的产品经理马化腾所提出的“运营驱动XX”的策略一样，依托运营团队，完成将服务推向市场，并根据市场用户反馈，以用户的体验为第一出发点进行产品改进，使反馈到产品优化再到用户反馈成为一个良性的生态循环系统。

■ 如何收集反馈

对用户反馈的收集主要有四种途径：第一种是通过各大网站论坛等用户公开交流及发布信息的平台，收集用户对于该产品的意见和建议；第二种是通过灰度发布的方式，令使用新旧两种产品的用户同时存在，根据用户的反馈效果来决定产品是否可以大范围发布；第三种是通过UE的用户使用调查分析，了解用户对于产品的反馈；第四种是通过在产品内部植入一些可以收集用户

反馈的数据桩，根据用户的使用情况具体分析用户的行为心理，进而了解用户对产品的反馈信息。综合运用各种反馈收集方式，尽可能全面和高效地了解用户行为数据。

■ **注重用户关怀**

所谓的用户关怀，即是对用户反馈的反馈。对于用户的反馈信息，除了要做到及时的收集和处理，更要做出积极的响应，并让用户可以看到产品的不断优化进步。在产品上线后，运营便在产品的生命周期中上演至关重要的作用，而对用户的关怀和回馈，除了通过产品升级以及在升级中说明修复的内容外，最主要的还是要依赖于运营的力量，在各大论坛以及客服电话等平台对用户的反馈进行答复和回访。只有这样，才能拉近用户与产品的距离，从而构建良性生态循环，迭代推动产品演进。

不可小觑的软实力

对于产品经理而言，除了上述工作流程及执行中所必须的能力外，还要具备一定的软实力，这样才能在工作中事半功倍。

沟通能力

这里所说的沟通能力，并不是要求产品经理需要多么健谈，言语需要多么滔滔不绝，而是说在沟通中，需要具备逻辑正确性以及分析辨别优先级别的能力。

在管理中，沟通是一项极其重要的工具，尤其是对于产品开发这种多团队协作、甚至是异地团队协作的管理工作，如果没有良好的沟通，很难想象项目会成为怎样的一团乱麻。而产品经理会每天面对无数的需求和任务，如果不能很好地识别优先级，势必会造成忙死了还做不成事的惨剧。

■ **逻辑正确性**

这里所说的逻辑正确性，是说产品经理在进行沟通时，一定要先把事情的逻辑关系梳理清晰，所有的结论和安排都要经得起推敲。产品经理是一个项目的中心和灵魂，所有的开发、设计和相关人员均会以他的要求作为自己工作的标准。因此，所说的话、所做的决定，一定不能够前后矛盾，不能够反复修改，否则在项目中，势必会出现信息错乱的局面。

■ **分析优先级**

对于产品经理，可能每天都会收到来自各个方面的需求和需要做的事情，但在既定的环境下，相关人员的精力和所能完成的工作都是有限的。这就要求产品经理需要具备一定的分析能力，在沟通中，分析事态的优先级别，做到合理的安排各项工作。

事业心和责任感

成功没有捷径，无论做什么工作，想要成功，必定需要经历一个破茧成蝶的痛苦煎熬的过程。对于互联网产品经理而言，工作中所要面对的问题是繁多、琐碎的，如何来支撑自己不断地坚持下去，唯有对成功的强烈渴望和责任感，并相信服务可以改变世界。正如比尔·盖茨在创建微软公司时，他的梦想是让每一个家庭和办公室都拥有PC，如今，Windows已经是非常普及；而对于腾讯，从1998年创建到现在，短短的13年时间，几乎每个中国人的电脑桌面上都有一只QQ，它的出现，彻底改变了中国人的交流方式。

因此，作为产品经理，不要吝啬自己的想象，要相信自己所在从事的工作，相信这个世界可以为你的服务所改变，这是自己的事业也是自己的责任！



刘舒

曾任职于微软、腾讯和阿里巴巴等公司，从事产品和项目管理工作。现任联想移动互联及数字家庭集团高级产品经理，负责联想移动应用商店服务平台的产品及研发工作。



张婷婷

毕业于四川大学工商管理学院，现任联想移动互联及数字家庭集团助理产品经理，参与联想移动应用商店服务平台的产品需求分析工作。

【编者按】产品经理在工作中遇到的最现实问题就是缺乏合手工具和实用方法的支持。听说《YES！产品经理》的主人公阿泡正在做这个工作，因此，从本篇开始，我们将连载《阿泡的产品管理工具包》系列，希望能为大家带来帮助。

阿泡的产品管理工具包之产品经理的34个感想

文 / 阿泡

“对于产品经理而言，别以为一个产品上市了就算大功告成，可以马放南山了，这只能说是产品管理过程中新产品项目开发阶段的结束，其实重头戏才刚刚开始，我们还要依照既定的年度营销策略来为产品的市场价值交换呕心沥血。”

以上内容是我在公司IMPlayer上市后，产品部内部总结会上做的发言，其实对于我来说，说不说这话都和我现在的工作状态没有太大关系，因为我负责的V产品线的新产品还远在规划阶段，这些话是我说给亮仔和谢韵琳的，毕竟他们算是新人，让他们思想上有个重视也好。

“各位，虽然IMPlayer成功上市了，正如阿泡说的，这才是刚刚开始，咱们今天的会议，一方面是总结一下这段时间的工作，另一方面是安排一下接下来的工作，” 周扬还是那样不紧不慢地抽着烟说道。

亮仔摸了摸越发闪亮的脑袋说：“扬哥，你就放心吧，接下来的工作我和小谢一定会再接再厉，不辱使命。说实话，别看我是研发转过来的，但必须承认我做产品经理确实有一种天才般的感觉，就说那次吧……”

自从IMPlayer成功上市后，亮仔的气场就更强了，走到哪里都给人一种打了鸡血的感觉，按照他的话说，他现在也算是朝成功的产品经理更进一步了。

在我们BS的眼光中他依然不以为耻，反以为荣，没事就在市场部和研发部转悠，并恬不知耻地告诉这些团队中的人，IMPlayer有啥情况直接找他就可以了。

亮仔清清嗓子，看来是准备把内部总结会变成个人事迹报告会了，周扬扔给他一支烟，意思是赶快闭嘴，现在还没到论功行赏的时候呢。

亮仔果然是禁不起糖衣炮弹的攻击，在这支烟的诱惑下，他果断地放弃了继续个人事迹报告会的意图，等着周扬安排工作。

周扬又拿出两支烟，一支给我，一支自己夹在手指间，但没有点上，旁边的小娟提醒我们三个烟民这已经是各位本次会议的最后一支烟了。

周扬边在指头间拨弄烟边说道：“其实这也是公司的安排，公司要转型，要更加正规，很多工作必须现在就重视并开展起来，对于咱们产品部来说，就是要逐步建立起规范的产品管理知识体系出来，也算是公司知识管理的一个组成部分吧。对了，泡，你对构建公司这个产品管理知识体系有什么体会或者经验没有？”

我挠挠头，说道：“这个还真没什么太多经验，以前的公司也没太重视过这个工作的开展。不过既然是公司要求建立这个知识体系，那咱们还是老规矩，边琢磨边实践吧，我个人觉得现在可以做的就是把前段时间的工作完全

梳理一下，至少可以把一些原则性的东西和工作中涉及到工具方法什么的总结出来，我觉得这是比较现实和可行的。”

“好，就按你说的，先从总结工具方法入手，这段时间咱们也摸索了一些思路和方法，花点时间把这些形成一个体系和流程。不错，谁来做呢？”周扬自言自语道：“亮仔和小谢得跟进IMPlayer的上市情况，肯定没什么时间。小娟和苏勇经验不行，现在就是阿泡和刘宇了，你俩看谁承担一下这个工作呢？”

没等我表态，周扬就对我说道：“我看这样吧，就由你来负责吧。”

说完，就让小娟把决议记到会议记录中，我赶忙说道：“别呀，我也很忙的，我看是不是再考虑一下，让大家都发表一下意见……”

“嗯，你说得对，咱们产品部是该讲民主的。这样吧，大家举手表决，举手的表示同意，不举手的表示无疑义。”

周扬话音刚落，“唰唰”就举起六只手，周扬一乐，说道：“好，全票通过，这个工作由阿泡来完成。娟子，把这次民主表决的结果记录下来，以备查阅。”

顿时，我只觉得天昏地暗，产品部竟然这样暗无天日啊，头脑中不由地涌现出一个做产品经理的老外朋友说的两句话：

A Business Is Not a Democracy（商业无民主可言）

Product Management is Inherently Political（产品管理就是内部政治）

当时我还不信，现在来看，真是至理名言啊！

改变周扬想法的可能性是几乎没有了，但我还是要争取一下我应该有的权利，于是我说道：“既然大家这样黑我，那我就认了，不过我有一个条件。”

周扬终于点上了最后一支烟，慢悠悠地说道：“只要不要钱，不要人，什么都行。”

“那，那我没条件了！”

就这样，我的最后一点权利在周扬的色厉内荏中也化为乌有了。

“好，今天的总结会就到这里，娟子，把会议记录整理一下，CC给韩老大一份，散会！”

整个事情的经过就是这样的，不过周扬在会后和我说了，这个工作是一个持续进行的事情，没有明确完成的时间，只要认真做好就可以了。

既然周扬都这么说了，那我也就踏实多了，不过我和周扬说了，既然交给我来做，那具体怎么做就是我的事情了，周扬说那是自然。

于是，我就先总结了一些我对于产品管理工作的感想，其实在我看来，我是希望这些感想能够成为产品部的原则的。

以下就是我的一些心得。

1 我们为谁做产品决定了我们要做什么样的产品，我们做什么样的产品决定了我们的产品能走多远，我们的产品能走多远才最终决定了我们的企业能走多远。

2 在产品团队中，产品经理不是最聪明的人，最聪明的人在你的团队中，去发现这些聪明的人、聪明的想法。

3 产品经理想做好工作，必需要具备“五个一”：一个脑袋、一双眼睛、一对耳朵、一双手以及一双脚。

4 主动是要找到目标，好学是要实现目标，对产品经理这个人是如此，对产品经理的工作更是如此。

5 对于“产品”的认识决定了产品经理工作的广度，对于“需求”的认识决定了产品经理工作的深度。

6 对于产品经理来说，根本要关注的是如何提供给用户解决问题的最好方法，而不是提供给用户一个完美的产品。

7 一个好产品并不一定是一个能挣钱的产品，而能挣钱的产品，对于企业来说，则肯定是一个好产品。

8 作为一个产品经理，如果你还做不到为企业挣钱，那么就要先做到为企业省钱。

9 用户可以影响我们的产品方向，但不能决定我的产品方向。

10 教育用户这个问题的根本不是是否需要教育用户，而是教育用户的成本到底由谁来承担，这绝不能成为产品经理的一种惯性思维，一定视情况而论。

11 产品经理的理想且终极的目标就是：在合适的时间、合适的地点向合适的用户

推出合适的产品！

12 产品经理，一个在跷跷板上工作的人，一边是用户的利益，一边是企业的利益，做好了是左右逢源，做不好就是左右为难！

13 一个合格的产品经理应该是拉出去可以冲锋陷阵，拉回来可以排兵布阵，既有参谋的眼光，又有指挥官的果断！

14 一个产品经理需要了解的技术领域，宽度需要两英里，深度需要两英寸！

15 产品经理无非就做两件事情：知道做什么、知道怎么做。前者就是产品经理的战略规划，后者就是产品经理的战术指导。

16 产品经理，要相信自己的双腿，不要相信自己的耳朵！

17 作为一个产品经理，不要向别人轻易地承诺，也不要相信别人轻易的承诺。

18 情感是热血的，商业是冷血的，产品经理应该很多时候是一个凭热血去做冷血事情的角色。

19 产品经理在公司内的关系很简单，要么去和稀泥，要么去得罪人，但现在看，得罪人要比和稀泥更有前途。

20 都说忠言逆耳利于行，但对于产品经理来说，不但要有忠言，关键还得顺耳，才有可能利于行，这或许就是产品经理沟通能力的最高体现吧。

21 作为一个产品经理，在决策建议上可以允许失误，但绝对不能允许愚蠢。

22 成功就是不打折扣的执行，失败则是一个折扣接着又一个折扣的执行，因此，产品经理是不允许在业务执行上打折扣的。

23 无数的事实已经证明，一个产品的成功并不是由这个产品经理做对了多少决定的，而是由他的竞争对手做错了多少决定的。

24 产品经理一定是一个动脑比动腿多的职位，但要想更好地动脑，前提是腿必须先动起来。

25 这个世界上并不是没有需求，而是我们缺乏发现需求的眼睛。

26 都说产品经理的第一能力就是沟通交流的能力，怎么沟通、怎么交流，不是说你把该说的说了就可以了，而应该是把该说的说好了、说对了，说到团队成员的心窝子里才能证明你会沟通、会交流。

27 产品经理要面对的各类角色太多，很多时候都会形成一种对立面，在不涉及原则的情况，产品经理要学会先往后退。

28 产品经理一定要学会说“不”，面对的对象可能是我们的高层、伙伴以及客户，之所以这样，究其根本，是为了避免市场向我们说“不”。

29 对于产品经理来说，不是我们产品的机会越来越少，而是我们挖掘和发现机会的难度越来越大。

30 对于产品经理而言，要处理好各种关系，不但要以人为本，还要以人的利益为本。

31 对于产品经理而言，知道不做什么要比知道做什么更有价值。

32 把任何想法都记录下来，如果不记录下来，这个想法将永远只是“想法”。

33 对于产品经理而言，应该是没事不找事，有事不怕事。

34 产品经理最大的智慧就是把别人的智慧变成自己的智慧。

好了，感想就这些，大家认为是否正确那就是各人在实际工作中去体会了，不多说了，下班了，我得回去好好想想，这个工作怎样才能做好。



文章中所涉及到的人物、产品以及相关信息选自《YES！产品经理》一书，该书已由电子工业出版社博文视点出版。

打造高效的产品团队

文 / Marty Cagan　　译 / 孙洋 吉绚 潘希颖 唐丰能

打造团队最佳阵容

对职业球队来说，拥有最佳阵容，是达成赛季目标的关键。但本赛季的最佳阵容并不一定适用于下个赛季；在体魄和速度上，年轻球员会更胜一筹；进攻策略和赛季目标会发生改变；伤病情况和薪金约束都会影响球队阵容。

对于职业球队而言，提升专业技能和职业水准是场“持久战”，需要不断提升球员技能，让球员尝试不同位置，提升后备队员实力，选择球队队长，招募新球员，培养明星球员。

假设教练或领队要高薪引进一位新人来增强球队整体实力，可迫于薪金约束，不得不面临艰难抉择：要么和一位现有球员解约；要么集体降薪，为新球员腾出薪水空间；要么暂时搁置这项人才引进计划。如果教练患有选择恐惧症，没有采取任何行动，会有什么后果？球队老板发现教练不作为，教练肯定得下课；就算球队老板没发现异常，球队也会萎靡不振，长期出不了线，球迷也纷纷用脚投票，老板不高兴，后果很严重。

经营公司和管理球队在方法上并无二致。公司的预算就相当于球队的薪金约束，它由管理部门提出，并经过董事会审核。要确保经费能换来卓有成效的工作，我们还要时刻关注人才引进计划，用合适的薪资吸引来需要的人才。然而，多数管理者在技能、人才、团队的管理上乏善可陈，换句话说，他们没有完成好公司和股东交给他们的工作。

拒绝花时间提高球员水平的教练，肯定保不住饭碗。同样，无法保证业绩增长，无法打造最佳团队的管理者，也必定会被股东炒鱿鱼。

其实打造企业团队和管理职业球队在方法上极其类似，我们只是没有意识到罢了。要想把工作做好，我们必须用有限的薪水（董事会通过的预算）招募最优秀的人才；不断地评估队员工作、提升团队实力，确保每位人才都能在自己的领域中发挥最大的价值；此外，要不断招募新人，挖掘现有成员的潜能，让他们不断创造新的巅峰。

团队管理三要素：选材、培养、淘汰

接下来，我们继续探讨打造高效团队的技巧。这次，我们走出球场，去园艺世界看看。

没有哪位园丁乐意锄地、播种、祈祷风调雨顺、等待生根发芽。打造美丽花园的过程是辛苦的，打造高效团队的过程也不例外，你必须经历这段艰辛的旅程。

选材

花园种什么花好？选材很重要。不仅要选漂亮的品种，更要注重该品种和其他品种的搭配。要考虑该品种是否需要从土壤中汲取过多的养分，从而影响其他品种的生长；现有土壤的养分是否满足种子的生长需求？

在业绩高速增长的公司里工作，管理者们往往花费大量时间招聘新员工，却很少花时间对应聘者进行个体研究，也没怎么花时间总结过去招聘中遇到的问题。可是只有正视自己过去在招聘中犯下的错误并改正它们，才能在新一轮招聘中招到合适的人。考虑清楚你为什么要裁员？员工为什么要离职？通常，招聘者侧

重考察应聘者的技能，却忽视了其他的关键因素，例如与企业文化是否匹配，和团队合作起来是否融洽。

招聘合适的人，还需要从产能和质量的角度来重新审视企业需求。你是否真的需要一位新的工程师或者产品经理？还是说低效率意味着需要重新定义流程，增加基础工具或质检人员？

很多招聘者和前来面试的人交流了不到一小时就匆匆做出聘用决定。只有花尽可能多的时间去了解应聘者，才有可能在第一次招聘时就选对人。积极寻求帮助，让一些你信得过并且有丰富面试经验的人加入招聘团队，能有效地提高招聘成功率。联系应聘者离职前的经理和同事，留心提问，不放过任何背景调查中获得的信息。

培养

花时间培养团队就像在给花园施肥。不能因为时间不足，而忘记给团队“施肥”。

培养团队是为了培养人才，开发出好产品，为公司的股东创造价值。培养方式包括：培训、激励、提高方法与技术、调整福利待遇，以及所有能提高员工绩效的方式。

培养团队还意味着进行资源优化配置，将合适的人安排到合适的岗位。但是，如果一位员工多次调动岗位，就要考虑淘汰掉他了。

培养团队还需要提升队伍的整体水平，帮助他们取得更好的成绩。具有挑战又能实现的目标是高效团队的前进动力。作为团队的管理者，你的工作就是设置好这样的目标，使他们不断进步。

淘汰

在选材和栽培阶段需要投入大量精力，表现不佳的员工会浪费过多资源，阻碍整个队伍的发展。就像花园里的杂草会抢夺其他的作物的营养一样，在产品团队里，被窃取的养分就是宝贵的时间。培训表现不佳的员工，会牺牲掉宝贵的时间。

为花园除草的过程是痛苦的。大部分的管理者总是选择回避，不到万不得已，不会启动淘汰机制。

有些制度限制了公司的淘汰机制，只允许辞退不工作的员工（国家和地方都出台过此项规定）。这时，你必须找到合适的方式尽早淘汰掉那些会拖团队后腿的人。“快刀斩乱麻”有利于尽快找到合适的代替者，让团队工作走上正轨。

培养优秀的产品经理

我以前论述过产品经理在产品组织中的角色定位。不少产品经理向我反映，看过这篇文章后，他们突然意识到自己在打造高效团队的工作上投入的精力远远不足，希望我分享一些打造高效团队的具体方法。

我希望每一位产品经理都能认识到打造高效团队的紧迫性与重要性。公司需要高效的产品团队。作为产品经理，如果你没有塑造好团队，没能为队员提供成长机会，人才自会流失。我一直坚信那句老话：“员工因公司而入职，因上司而离职”。

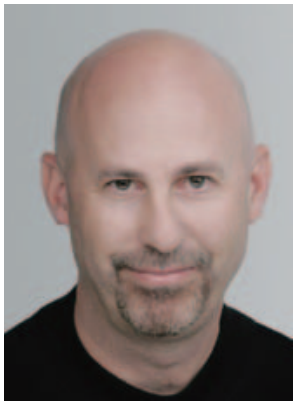
这里向大家推荐一种用于持续进行技能评估和人才培养的方法，这个方法我自己也在使用。一直以来，我把它看成一种团队培养模式。而有一些公司只在团队成员遇到问题需要改进时，才会采用这种方法。此举虽不为错，但我认为使用该方法的目的不应该是“亡羊补牢”，而是帮助培养人才，培养优秀的产品经理，持续打造高效的产品团队。

该技能评估和人才培养方式以“差距分析”为基础，用于明确员工需要在哪些方面进行提高，提高到什么程度。该方法有一系列评分标准，每个标准由两项组成，一项是目标分数，是指对于某一技能，员工需要达到的水平（也代表技能的重要程度）；另一项是员工目前对该技能掌握的程度（也就是员工的能力）。两项相比，差距一目了然。

并非所有的技能都一样重要，也不是所有差距都亟待消除，但它能帮助明确需要改进的地方。

技能评估

我把技能分为“知识”、“过程处理能



Marty Cagan

过去20年，Marty Cagan作为负责定义和开发产品的高级经理人为多家一流企业工作过，包括惠普、网景通信、美国在线、eBay。他亲历了个人电脑、互联网、电子商务的起落沉浮，致力于通过写作、演讲、培训帮助客户打造富有创意的产品。为此，他撰写了《Inspired: How to Create Products Customers Love》一书，创办了硅谷产品集团公司（SVPG）。在此之前，他的最后一份工作是担任eBay产品管理及设计高级副总裁，负责规划全球电子商务网站的产品和服务。

力”和“个人能力”三个维度，并在下文中对每个维度进行了简要描述。

这里我就典型中级产品经理所需具备的各项技能给出了目标分数（也代表重要性）。读者可以根据自己公司的文化、所处行业、团队、产品，以及职位（例如：对高级产品经理的要求比产品经理助理的要求高得多）对目标分数进行调整。

在等级评分中，0代表不重要（要求掌握的程度低），10代表非常重要（要求掌握的程度高）。

知识：

用户/客户知识（10）——产品经理在目标用户/客户方面是否是公司认可的专家？

行业/领域知识（10）——产品经理对该行业/领域的知识了解多少？

> 要想把工作做好，我们必须用有限的薪水招募最优秀的人才；不断地评估队员工作、提升团队实力，确保每位人才都能在自己的领域中发挥最大的价值；此外，要不断招募新人，挖掘现有成员的潜能，让他们不断创造新的巅峰。

产品知识（10）——产品经理的产品知识水平如何？

技术知识（8）——产品经理对基础技术了解多少？是否在与时俱进？

用户体验设计知识（7）——产品经理是否具备用户体验设计的相关知识？是否清楚用户体验团队需要具备的能力？是否足够重视这个团队的工作并最大限度地协助他们做到最好？

经营和财务知识（7）——产品经理是否明白产品的经营和财务动态？

过程管理能力：

客户探索过程（7）——客户探索包括客户面谈技巧、机会评估能力和对客户发展计划的理解。

产品探索过程（9）——该过程旨在打造出基本产品，既包括定性方法（如用户原型和用户测试），又包括定量方法（如实时数据原型和分离测试）。

产品优化过程（9）——这个过程要求产品经理具备快速提高和改善现有产品的能力，尤其是优化技术和A/B测试。

产品开发过程（7）——产品经理是否对产品开发过程（如Scrum开发方式）有深刻的认识？是否明白自己在创建和管理待开发任务中的重要职责？

个人能力：

团队合作能力（10）——产品经理如何与首席工程师及设计师一起高效工作？他们之间是合作关系吗？是否互相尊重？产品经理是否做到尽早让首席工程师和设计师加入产品团队，并为他们提供直接面对客户的机会？

产品推广能力（9）——产品经理能否有效地向整个产品团队、股东及公司其他员工展示产品愿景，说服他们为产品发展做出贡献？

时间管理能力（8）——产品经理是否能管理好自己的时间？宝贵的时间是用在了“刀刃”上，还是耗费在日常紧急事务上？他有没有让团队里的项目经理或Scrum专家充分发挥作用？

与股东的协调能力（7）——在与股东的协调过程中产品经理的表现如何？股东们能否觉得产品经理是值得信赖的商业伙伴？

领导能力（6）——事实上产品经理并不真正管人，但是他们确实需要去领导、影响并激励员工，因此领导能力对他们来说很重要。

用户管理能力（5）——在用户管理以及平滑部署上产品经理会采取哪些方法？

产品的整体观（5）——产品经理能否保持产品的整体观，是否有能力为用户提供由始至终的产品体验（end-to-end experience）？

设定完成每个维度的目标分数后，就可以对产品经理进行评分了，从而找出与目标之间的差距。

培养计划

技能评估仅是第一个步骤。接下来要找出产品经理差距最大，最为薄弱的环节，并据此制订一系列培养方案。比如“产品探索过程”的目标分数是9，而某位产品经理目前的得分是

5，那么他还需要在这个重要的技能的培养上多下工夫。作为管理者，你应该给这位产品经理提供培训、阅读或者实践项目来提高他在产品探索上的技能。

培养计划要有的放矢、逐步进行，先集中攻克三个差距最大的技能。产品经理在这三项技能上取得进步后，再启动下一阶段的培养计划。

一旦成功地消灭了这些差距，便是时候关注更高一级职位（或其他职位）的技能目标，以便开始着手培养晋升所需要的技能。

要经常跟进产品经理的培养进度，调整目前的技能评分，做到每月至少一次。能力特别强的产品经理则需要每周跟进一次。

也许，你不明白技能评估和培养计划与绩效评估之间有何联系。我发现大多数公司的绩效评估对打造高效团队起不到任何作用。绩效评估只和员工是否遵守制度以及薪酬有关。

也许你不得不配合HR部门的绩效评估，但绩效评估不能积极、持续地培养团队成员的技能。你可以选择技能评估和培养计划，如果运用得当，绩效评估对你来说就是小菜一碟。

推倒产品团队里的那堵“墙”

创业公司的最大优势就是规模小，无所谓组织结构、无所谓分工。所有的员工只有一件事要做——打造人们想要的产品。

随着公司规模不断增长，公司的注意力慢慢从“打造产品”转向了人员和流程的管理，职能开始被细分成三大类：技术、产品和市场营销。

于是，原本一体的公司，被一面面无形的“墙壁”隔离开来。相较于自己实际工作的团队，员工们更愿意亲近和自己职务相近的那群人。观察下和他们一块儿用餐，一起喝点儿小酒的人，就不难发现这点。如果产品经理只和产品经理关系好，那堵“墙”就产生了。

如果产品经理和开发人员无法达成共识，或者不了解对方的工作，显然他们之间也存在着那堵“墙”。

而Scrum开发方式能推倒那堵阻碍团队合作的“墙”，因此深得我心。但即便拥有Scrum团队，产品经理也不一定是真正的产品负责人，或者产品经理只是名义上的产品负责人，却未真正参与其中。

当然，还存在很多其他无形的“墙壁”阻碍着团队的沟通与合作，比如屹立于产品部门和市场部门之间，或是产品和用户体验之间的那堵“墙”。但产品和技术之间的那堵墙最为严重。如果开发团队和产品经理无法进行有效的协同合作，到头来只会一事无成。

对我而言，有效的协同合作是建立在私交之上的。

好的产品是产品团队、用户体验团队和技术团队之间精诚合作的结晶，而精诚合作得益于他们之间的私交。

推倒墙壁的方法很简单：产品经理只需要把办公桌挪到开发人员的旁边，就能产生明显的效果。很快，他们开始一起进餐，一起玩桌上足球，直至一起探讨观点。

我在视频采访中提到过产品团队的三大重要角色（产品经理、交互设计师、主程序员）应该如桃园结义的三兄弟般亲密无间，他们应该在一个“桃园”里办公。【注：视频地址：http://v.youku.com/v_show/id_XMzA4MTU1NzI0.html】

在别的文章中，我还提到只让开发人员负责软件的开发，明显低估了他们的价值。如果想让开发人员充分展露自身能力，就尝试一下上述方法吧！



本文节选自华中科技大学出版社《启示录：打造用户喜爱的产品》一书及作者的博客。该书从人员、流程、产品三个角度介绍了现代软件（互联网）产品管理的实践经验和理念。特此感谢华中科技大学出版社与Marty Cagan先生授权。

产品管理，明天会如何发展？

文 / 张智渊

随着产品管理在国内的迅速发展以及相关体系的不断完善，在接下来的发展过程中，产品管理会呈现怎样的发展特点呢？本文将对产品管理在国内的发展方向、重点发展行业以及可能形成的市场规模进行预测。

发展依旧不平衡

产品管理的发展一直呈现出行业发展不平衡、地区发展不平衡、企业发展不平衡三个最突出的特点，这三个特点不会在短期内发生改变，而是会依旧呈现出来。

之所以出现这种情况，一方面是因为产品管理在国内的发展，缺乏足够的推动力，更重要的一方面是因为行业差异、地区经济水平以及企业认识的差距造成的。

在国内，产品管理发展较快的行业主要集中在新兴高科技行业、与国际市场有较多接触的行业，以及传统的采用产品管理体系的行业，主要集中在以下几个行业。

- IT行业：软件、互联网、硬件、通信。
- 金融行业：银行、证券公司、保险公司。
- 快消行业：食品、日化用品、保健品。
- 医药行业：生物制药、医疗器械。
- 家电企业：家用电器、工用电气。

而对于那些传统生产企业，则发展比较缓慢。不过，在接下来的3~5年内，相信将会有越来越多的传统行业采用产品管理。之所以这么说，原因有三。

中国市场融入国际市场的步伐越来越快

现在，市场的界限越来越模糊，大量的国外企业进入国内，同时，大量的国内企业也开始走出国门。这时中国会发现，参与国际市场的竞争，不单是简单的产品、技术层面上的竞争，而越来越多的是企业本身资源的综合较量，其中非常重要的一项就是如何建立一个运转高效、反应快速的企业资源流转体系。当然，这种体系不仅只有产品管理一种，但产品管理确实是一种非常有竞争力的体系，相信会有越来越多的企业愿意尝试使用这种新型的管理体系。

市场竞争越来越国际化、多元化和白热化

由国内市场竞争转到全球竞争，这不是由企业意志来决定的，封闭的国内竞争已经结束，要想在长期的市场竞争中站稳脚跟，国内企业就不能再用短期的眼光来审视市场，而是需要用长期的、发展的眼光来规划企业的发展。

而产品管理有一个明显的特点就是产品战略、产品规划和产品战术三位一体紧密地结合，产品管理者一方面要为产品制定长期（3~5年）的发展战略和阶段规划，另一方面还要作为产品的唯一负责人来监控战略的实行，这样才能保证在产品的长期发展中，有专人进行专业的、有针对性的管理，从而保证企业产品能够按照既定的步骤来执行，增强企业产品的市场竞争能力。

企业开始追求“性价比”高的生产流程

企业的资源毕竟是有限的，但面对市场竞争时，如何才能把有限的资源投入到能够带来最大收益的市场中呢？这就给所有的企业提出了一个最现实的问题：企业应该有一个“高性价比”的产品生产流程来保障。

从产品战略开始，一直到产品上市，在整个过程中，如何能够保证企业资源是处于最大化的应用，产品管理应该是最有效的保障了。

产品管理能够在产品处于构想时，就保证企业资源进入到一个正确的过程中，而不会出现资源使用的随意性。

因此，当企业面临这些问题时，就需要找到

一种适合企业长期、有效参与市场竞争的体系来进行运作，而产品管理则是最具魅力的一种。

由此可以看出，在3~5年内，产品管理在国内传统行业中，必然会被越来越多的企业所采用。但这不可能一蹴而就、一步到位，认识的提高和体系的完善需要一个长期的过程，有发展，但不会太快。相比较那些已经采用了产品管理或者易于采用产品管理的行业，差距还是非常明显的。因此，发展的不平衡在短期内会持续存在。

逐渐形成符合国情的独立知识体系

这是毋庸置疑的，我们知道，任何理论体系的建立，都是基于大量的实践之上的。产品管理基于Marketing，但随着Marketing自身的变化以及更多企业对产品管理现实需求的不断加大，现在越来越多的迹象表明产品管理开始逐渐形成一门独立的学科：产品管理学。

在产品管理进入中国到目前为止，已经有不少的企业进行了尝试，但通常都是照搬国外产品管理的一些皮毛而已，在我看来，这个过程是必须的也是非常有意义的。

- 由模仿到自主，对于一个基础薄弱的学科来说，是最佳的方式，一方面成本低，另一方面可以缩短意识差距。
- 在这个过程中可以积累大量的实践案例，而这些案例则是形成符合国情的产品管理的宝贵财富。
- 可以发现国外产品管理思想中的不足，产品管理是舶来品，所处的社会、市场环境的不同决定了这个思想被应用到国内企业身上时，很容易发生“水土不服”的情况，通过模仿学习，可以发现国外思想中的不足，然后结合国情加以改造。

国外产品管理发展了80年，国内发展也不过十几年的时间，在这段时间中，采用了产品管理的国内企业产生了大量案例，而这些案例将为形成符合国情、独立的产品管理思想奠定基础。

因此，在产品管理的后续发展中，肯定会有越来越多的社会机构和个人对这些案例进行整理、分析和总结，开始形成自己的产品管理思想体系，出现“百家争鸣”的情况。

在我看来，从“照搬”到“自主发展”再到“百家争鸣”，将是5年内必然出现的情况，而这种情况的出现，对于推动国内产品管理的发展非常有利，但需要注意的一点是：现在不少开始自主发展的机构和个人，因为思维的惯性，需要一段时间从国外的知识体系中脱离出来，而在这个过程中，肯定会出现“大杂烩”的情况，而对于意欲采用产品管理的企业来说，这是需要非常慎重的。

因此，在接下来的5年内，是一个逐渐形成符合国情的、独立产品管理学科的阶段，而在这个阶段中，必然会是“百家争鸣”，并不断地和国外的产品管理思想进行互动，最终形成符合中国国情的产品管理思想体系。

行业差异明显，产品管理出现细分

这里所说的行业差异，主要是指“不同行业对产品管理的认知度会因为行业的不同而有所差别”。这个特点和第二点紧密关系，产品管理发展最快的三个行业——IT、医药和快消——各自对产品管理的认识都相差很大。

即使在IT行业，软件、互联网、硬件和通信这四个具体行业，对产品管理的认识也是有所差异的，其根本是由这些行业的产品特点决定的。

提供服务性产品的行业和提供实物类产品的行业，其对产品管理的侧重就完全不同，例如在互联网行业，强调的是对产品本身的需求分析和产品定义；而在快速消费品行业，则更强调的是对产品在市场上营销资源的控制。

我们知道，现在出现的许多行业，在国外产品管理形成阶段是还没有出现的，因此，许多新兴行业在采用产品管理时，就不能直接采用产品管理的体系和流程，一方面要吸收国外产品管理的精华思想，另一方面还要在此基础上加以改造和发展。

在第二点中，提到在接下来的5年中，会逐步形成符合国情的、独立的产品管理思想和知识体系。同时，在这个形成过程中，会因为行业的差异性，很有可能会细分产品管理知识体系，例如IT行业产品管理知识体系，金融行业产品管理知识体系等，而这些细分体系的形

成，除了需要社会机构的参与外，更重要的是需要大量的行业从业人员的参与和支持。

但并不是说，这种细分的行业体系会脱离于第二点中提到的独立的产品管理知识体系而存在，因为这两者之间的关系就如同水和鱼，产品管理知识体系如水，是鱼赖以生存的环境；而细分的行业体系则是鱼，必须在水里才能生存下去，无论走得多远，都不会脱离开水。

我认为的理想状态是，在独立的产品管理体系下，有大量熟悉本行业的产品管理专家来研究行业产品管理知识体系的发展。

以上三点预测了产品管理在国内未来发展的特点和趋势，总结一下，就是会出现“两极分化、本土化、差异化”的情况。

未来可能大规模采用PMS的行业

接下来，我对将来可能会大量采用产品管理的企业类型进行一下预测，这里不涉及到具体的行业，因为接下来所提到的企业类型在各个行业中都有存在。

通过分析，我认为在未来的3~5年内，产品管理会在以下三种类型的企业中快速发展。

- 由项目型转型为产品型的企业。
- 由提供单一销售向提供解决方案转型的企业。
- 由单一制造向完整市场运作转型的企业。

比如在金融行业的个人理财产品上，在未来的发展中就面临着由单一销售向为个人用户提供个人理财解决方案的要求。

这是因为随着国内经济的不断发展和个人收入的不断提高，个人理财已经成为一个非常热门的主题，而在这个主题背后隐藏着的需求就是“个人理财产品的单一型”。现在，有许多金融企业都已经看到了这个潜在的需求，纷纷推出多样化的金融服务，包括银行、保险公司、证券公司等，都已经开始推出多样化和个性化的金融产品，例如现在比较热的基金产品，已经多达上千种。同时随着国外金融机构的进入和人民币市场的逐渐开放，国内金融企业的压力越来越大，如何才能应对这种挑战，主要的一点就是要细分金融市场的用户，而不能像以前一样，只是简单地分为企业和个人用户，尤其在个人用户市场中，要求细分的需求更加强烈。

因此，在金融市场，个人要求产品细分和提供完整解决方案的需求非常强，这就要求金融企业必须建立产品管理来保障需求的实现，在未来的5年内，金融行业的产品管理必将会有迅猛的发展。

同样，在家电行业也是如此，在经过了“战国时代”后，家电行业在不同的领域已经形成了“三国鼎立”的市场局面，也就是说，家电行业中的资源已经由分散开始集中到几家大企业当中，并且，这些家电企业已经开始走出国门，开始参与国际竞争。

如何利用这些资源去赢得市场，就是摆在每个家电企业面前的最实际问题，在产品管理中，就不止一次地强调“产品管理一定是要考虑全球化的市场战略”，而这正好为国内家电企业的国际化发展提供了发展支持。

这就是由单一制造向完整市场运作转型的企业所必须面对的前景。

在其他行业内，企业发展很不平衡，有基础好、发展快的，有基础差、发展慢的，具体何种企业适合采用产品管理，我们会另文再说。而我当然希望越来越多的国内企业采用产品管理。

结束语

在本文中，我们对产品管理在未来3~5年的发展进行了预测。既然是预测，那就不妨大胆一些，相信产品管理的发展，不是谁能够主导的，当然，也不是谁能够阻挡的，这是由客观的市场发展趋势决定的，不论是企业、个人还是提供产品管理服务的机构，唯一要做的就是抓住机会，发展自己。



张智渊

UCPM中国产品经理联盟发起人，长期关注中国产品管理体系的建设和发展，曾在国内多家知名企业从事产品管理工作，具有丰富的一线产品管理经验和扎实的理论基础。

技术人员如何转型为产品经理

技术人员做到一定程度，有些会继续在技术和技术管理的道路上走下去，成为技术经理、技术总监直到CTO、技术总裁；另外一些会想到要转型，换到其他岗位上继续自己的事业，产品经理成为他们的一个好选择。但从技术人员转变为产品经理，不是一朝一夕、轻而易举就能成功的。本专题邀请几位有转型经验的曾经的技术人员，分享他们的转型秘诀。



王晖
金山办公软件移动办公部总监

从“if”和“else”到“也许”

在成为一名产品经理之前，首先要搞清楚产品经理要做些什么工作。一个我可以接受的描述是：产品经理其实是这个产品的总负责人或总负责人的继任人。在一个新产品的团队中，他应该负责的事情至少包含：产品定位、运营策略、销售政策、产品发展规划。能够做到这些，他和总负责人相比几乎就差“人员管理”和“进度管理”了。由此看来，产品经理并不是一个很多人以为的“退而求其次”的角色。

我能成为金山Office移动版的产品经理，直接和我在Office软件开发领域中的经验积累相关。从2000年~2010年，我的主要成长路线是：程序员→开发组长→开发经理→项目经理→研发部技术副总监→研发部技术总监→研发部副总经理。这些经历让我有了较多的技术基础和管理经验。

在担任研发部技术总监和研发部副总经理期间，我和研发部的产品部总监共同探讨了很多关于产品定位、营销、交互设计相关的问题，看了一些产品和设计相关的书籍和资料，并且推动这些知识在研发部内进行实践。我想，拥有这段时间的经历，是我能够担任金山 Office移动版的产品经理角色的重要原因。回顾这段时间，我最大的体会就是：技术人员要转型产品人员，首先要清楚地认识到，在工作中除了“if ”和“else”，还有一个“也许”。如果内心能够坦然接受“也许”带来的不确定性，把“我们边走边看吧”也作为一个心安理得的选项，那么，这应该是从技术人员转型成为产品人员的重要一步。

另外，我认为一般人要成为好的产品经理，交互设计这一课是必须要上的。时间长短不限，但不能不研究。就好像要做一名架构师，必须要会进行函数级的编码一样。从交互设计入手，可以由浅入深地了解什么产品是好的产品。如果跳过这一环节，直接谈定位、谈发展，很难避免进入胡夸、幻想的困境。一些人有机会直接成为产品经理，但我猜他们要想做好的话，肯定是会要回头补上这一课的。然后需要的就是随着时间而来的各种经验。

最后补充一点：技术人员能够坚持在程序开发技术和项目管理方面有所建树，也是一件光荣和有前途的事情。



刘波
南都全媒体•奥一网首席产品官

技术如何转型为产品经理我之见

一次茶余饭后跟技术中心的研发工程师天南地北地胡侃，不知不觉话题就转到了产品经理这个职业上。小吴说：“哎，做了将近四年的研发，越来越觉得没劲，还不如转做产品经理。产品经理多容易呀，动动嘴皮子，用RP画画图就行了。像我们做研发的，转做产品经理就更加容易了，可以无缝转型。”我听完一脸诧异，无语……

虽然产品经理职业早已存在，但在互联网行业却算新兴，也是当下最热门、最让人趋之若鹜的职业之一，是技术人员转型的首选。但很多技术人员普遍存在一个错误的观念，总认为自己懂技术，转做产品经理还不轻而易举？在我从技术转型做产品经理将近三年的时间里，我越来越觉得从技术人员转型为产品经理是一个巨大的转变。为何这样说呢？我们先来弄清楚下面三个问题。

■ 产品经理是什么？产品经理是个内涵非常丰富的职务，是一个产品从开发到运营贯穿始终的负责人，是项目组的润滑剂，是技术与市场的交汇点。

■ 产品经理需要有什么？我认为，产品经理首先需要敏锐的市场触觉，其次需要良好的沟通能力，再次需要有较好的项目管理能力，最后还需要有较强的数据分析能力。要成为一个好的产品经理，这四者缺一不可。行业内顶尖的产品经理都具有很好的对机会的嗅觉和洞悉市场变化的能力，通常重视用户、数据和事物的细节。

■ 技术人员通常具有什么特点？技术人员的普遍特点是有毅力、喜欢思考、逻辑性强、喜欢钻研，但往往沟通能力不强、敏锐度不够、容易钻牛角尖、视野不够开阔。

由此可见，技术人员在转型为产品经理的过程中，由于自身的特性会遇到一些瓶颈。因此要想成功地从技术转型为产品，我认为可从下面几个方面着手。

■ 在决定转型前，要先全面了解产品经理的工作职责。

很多人都觉得产品经理还不容易？整天就看到这帮人在上网、跟同事聊天、画画框架、写写文档，有什么难的。其实不然。产品经理是个既当爹又当妈的角色。产品经理既要承担项目的管理者、推动者的责任，又要肩负产品规划和设计的重任，同时产品上线后还要做好数据分析、市场反馈收集等后续工作。

■ 开始产品经理之旅后，需要尽快转变思维。

转为产品经理后，我们要摒弃凡事从技术可行性着手的惯性思维方式，要从“产品”走向“市场”，要从“技术”走向“业务”，从“单兵作战”走向“协同作战”。所有产品的产生都来源于市场需求，因此必须培养自己的市场触觉，多了解市场的业务信息，多收集行业趋势动态。

■ 要拓宽视野，立足全局规划产品。

产品经理所隶属的产品中心（或产品部）应当是一个“抬头看天”的部门，因此产品经理也必须要有Vision，不能只盯着脚下，要懂得抓大放小。

■ 要学会遗忘技术，把自己变成一个普通用户。

如何能够设计出用户体验好的产品呢？技术人员转型为产品经理还有很重要的一点就是要学会遗忘技术，把自己变成一个普通用户。现在互联网竞争异常激烈，往往细微的体验差异就会导致竞争力的强弱。而最好的产品体验就是傻瓜式的操作。因此我们要先把自己变成“技术白痴”，才能设计出操作简单、体验好的产品。

技术人员转型为产品经理是个巨大的转变，涉及到思维方式、工作方式、合作对象等方面。转型是个复杂的过程，但只要我们坚持，善于接受意见，懂得聆听，相信一定会看到曙光出现的那一天。



王正培
北京超图软件股份有限公司产品经理

不能只以研发人员的角度考虑问题

研发人员如何转型做好产品经理，是很多研发人员现在或者将来都可能会面临的问题，下面就分享一下我一年多来产品经理工作的点点滴滴。

初当产品经理

产品经理一般是由具有丰富开发经验的、对产品有着深刻理解的人员担任，当然有的产品经理也不一定是开发人员。因此在研发人员转型担任产品经理之初，难免会有一个相对痛苦的过程。以前在担任小组Master时每天只需要和小组成员明确工作内容，完成需求列表的功能，修改测试或者用户上报Bug，定期向上级经理汇报工作就可以。而做了产品经理之后，我们不仅需要了解团队每个人的工作，而且相关多个团队的工作进度都需要了解，毕竟用户面对的是整个产品体系，不是单独某个产品，同时需要支持中心、培训中心、各分公司事业部，以及一线用户的需求和反馈问题都要了解和跟踪。总体上说，产品经理的工作更繁杂了，需要有更强的责任感。

产品经理注意事项

■ 考虑问题需要更加全面，不能只以研发人员的角度考虑问题，同时也不能只考虑外界用户。既要考虑研发产品的周期性，也要考虑项目时间点和功能点，给项目实施的用户一些可行的建议，切忌不能过多吹嘘自己的产品，一定要实事求是地介绍产品功能。

■ 做好承上启下的传令兵。对于外界用户反馈产品好的方面，要经常传递到做研发的同事中，因为他们更多接触到的都是产品的Bug，很少听到对产品的赞美，多一些赞美可以给研发团队增加信心，保持战斗力；对于外界反馈的问题要采用辩证的方式反馈到研发中心，重要问题要及时跟踪并反馈，对一些研发的低级错误也要严格对待。

■ 对突发应急的需求保持时刻关注。用户永远希望他们发现的问题能在最短的时间得到解决，因此周末或者半夜接到咨询电话、出差到各个分公司支持项目会是一种常态。

■ 心态一定要积极。在对项目支持的过程中，难免有些项目因为功能缺陷或者发布时间问题导致项目流产，这时项目实施人员可能会对自己有投诉或者写邮件向上级领导反映，而产品经理一定要保持良好心态，反省整个事件发生的过程，检讨自己在支持过程的问题，避免下次再犯。此外，产品经理需要保持良好的心态去继续支持后面的项目，要知道用户反馈的问题越多，说明用的人越多，对我们完善产品越有帮助，这是具有积极意义的。

■ 积极主动。对于业界的最新动态时刻保持关注，对同类产品的动向时刻保持警惕，对于项目上积累的经验能产品化的尽量产品化，不断完善产品，降低产品的开发难度，对产品解读不遗余力，始终要把用户体验放在第一位。

产品经理在路上

最后，我想给自己一些展望，希望在今后的工作中更加完善自己。首先，需要进一步完善自己的知识体系，对企业所有产品都要了解，因为面对用户，你就是企业的代言人，能在你这里得到全面的反馈，就能够减少用户和公司的沟通成本，继而也使自己也能成长成为一位合格的产品专家；其次，需要深度挖掘各个行业用户对产品的需求，对每个行业整理出完善可行的解决方案，因为不管什么产品，用户说好才是真的好，行业应用做得越深入，同类产品就很难对我们产生威胁。

Justep X5产品揭秘

文 / 宋兴烈

Justep X5产品定位和应用效果

近年来，随着互联网、电子商务、移动技术的快速发展，企业信息化和管理软件领域也发展到了一个崭新的时期和前所未有的高度。如何在互联网和移动环境下构建企业的信息化系统，已经成为业内的关键问题。Justep X5的产品定位，归结为一句话，就是“互联网和移动技术下的信息化平台”。

在互联网和移动技术环境下全面实现企业信息化

Justep的基本目标，就是为基于互联网和移动环境的信息化提供强力支持，企业的信息化应用，往往涉及复杂的业务逻辑和人机交互，所以，必须对Web下的复杂业务系统有很好的支持。

对移动业务和便携设备提供全面支持

随着移动网络和移动设备的普及，对移动技术的支持，已经成为企业信息化的基本要求。X5应该对移动业务提供全面支持，对Android、iPhone和WinPhone三大移动设备均提供良好支持。

实现互联网和移动业务信息系统的一体化和资源重用

在移动和互联网络高度融合的背景下，必须确保这两类信息系统的一体化和资源重用。在JustepX5平台上，无论采用何种网络、何种手机前端和PC端，所有工作流、组织机构和权限、业务逻辑等服务端业务逻辑，都是可以重用的。

实现互联网和移动技术下的软件快速开发和灵活调整

X5是业内唯一的基于互联网和移动技术下的业务建模工具。X5业务建模是以业务描述而非代码为核心来构建信息系统， X5业务建模工具实现了业务导向和驱动的软件开发，从根本上提升了信息系统的开发和维护效率。

保证技术的开放性和标准性

Justep X5确保整个系统的技术开放性和标准性。在服务端，X5采用标准的Java和SOA技术；在客户端，X5采用浏览器和纯JavaScript技术。在集成开发环境上，X5采用Eclipse平台。此外，在工作流、业务逻辑、组织机构和权限、用户界面等各个方面，JustepX5几乎全部采用W3C的标准。

Justep X5业务建模体系

可执行模型驱动方法可以极大地提高生产力

可执行模型驱动方法就是在目标领域内，根据预先制定的适合该领域的模型规范，采用建模工具建立模型（即应用），通过模型运行环境执行模型，得到应用系统。

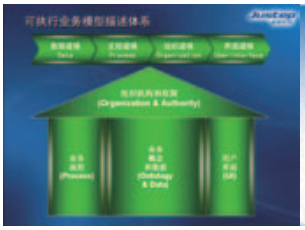
Web世界不仅丰富多彩，而且Web页面的生产力是如此之高，是什么使得Web页面制作的生产力如此之高呢？答案就是制作Web页面的方法是一种在超文本页面这个目标领域内的可执行模型驱动方法。HTML、CSS、Javascript是模型描述规范，Dreamweaver、Frontpage是遵循该规范的建模工具，Web服务器和Web浏览器是运行环境。让我们想象一下，如果Web页面不是采用可执行模型驱动这种方法，而是采用3GL语言编码的方式来制作页面，生产力能达到现在Web页面的生产力吗？会有今天如此丰富的Web世界吗？

针对某个领域的可执行模型驱动方法极大地提高了该领域应用开发的生产力，这种例子还有很多，比如Flash、嵌入式领域的SysML、工作流系统等。可执行模型驱动方法最重要的是可执行模型描述规范。找到和制定适合目标领域的模型描述规范是可执行模型驱动方法的关键。

Justep X5在信息化领域的可执行业务模型驱动方法

Justep X5是针对信息化领域的业务架构平台，采用可执行模型驱动方法来提高开发信息化应用的生产力。

Justep参照和遵循相应的国际标准，制定了适合描述信息化业务的模型描述规范体系。Justep的业务模型规范体系目前包括四个基本的模型视图：组织模型、流程模型、数据模型、用户界面模型。这四个模型视图可以涵盖信息化业务的主体需求。这四个模型视图的详细具体的描述规范都遵循或参照了相应的国际标准（如表）。



Justep业务模型描述标准体系

业务模型	描述标准	标准组织
业务数据模型	OWL	W3C
业务流程模型	BPMN	OMG
用户界面模型	XHTML、CSS、XForms、XBL、SVG、Javascript	W3C
组织机构模型	LDAP	IETF
业务权限模型	RBAC	NIST

W3C，万维网联盟　IETF，互联网工程任务组
OMG，对象管理组织　NIST，美国国家标准技术研究院

根据上述业务模型规范，Justep开发了相应的业务建模工具Justep X5 Studio，该建模工具是基于Eclipse集成开发环境的业务建模插件，

这样既可以方便和可视化地建立业务模型，也可以做业务模型的扩展代码开发。

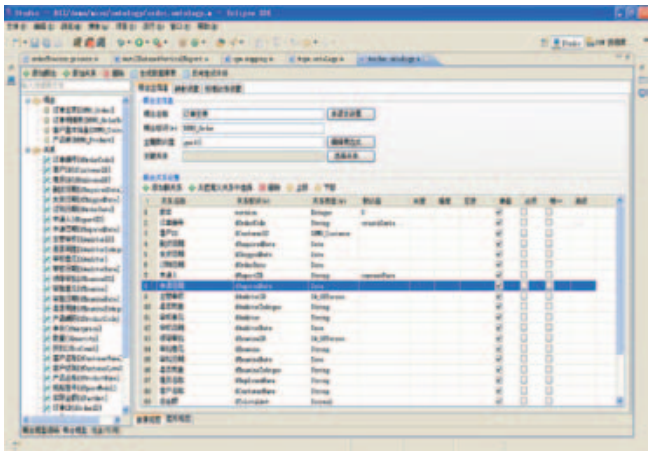
Justep也开发了基于该模型规范的运行环境Justep X5，可以很好地运行依据该模型规范建立的模型应用，得到相应的应用系统。运行环境是基于SOA的JavaEE B/S体系结构。组织模型、流程模型、数据模型在业务服务端运行，提供服务给外界调用，这些服务受权限模型的控制。而用户界面模型是放在Web服务器和浏览器端的，PC和Mobile各有自己的用户界面模型，但共用组织模型、流程模型、数据模型。

Justep X5通过业务模型描述规范、业务建模工具、业务运行环境，形成一套面向信息化领域的可执行模型驱动方法和工具，来极大地提高信息化应用开发的生产力，更好地满足信息化需求。

Justep X5可执行业务模型驱动方法的实际开发过程实践

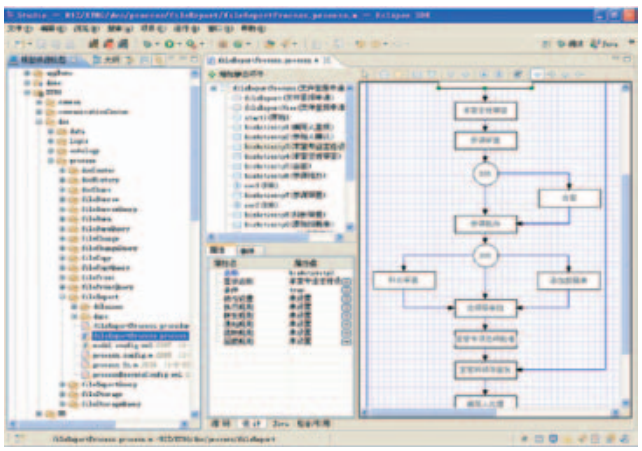
对应业务模型规范，通过X5建模开发信息化应用的步骤分为四步。

数据建模



数据建模包括建立概念数据模型、物理数据模型、操纵数据的标准动作，还可以通过Java扩展代码开发自定义业务逻辑动作。这些数据标准动作和自定义动作在模型运行时公布为模型的业务服务，供多种用户界面模型或者第三方调用。

流程建模



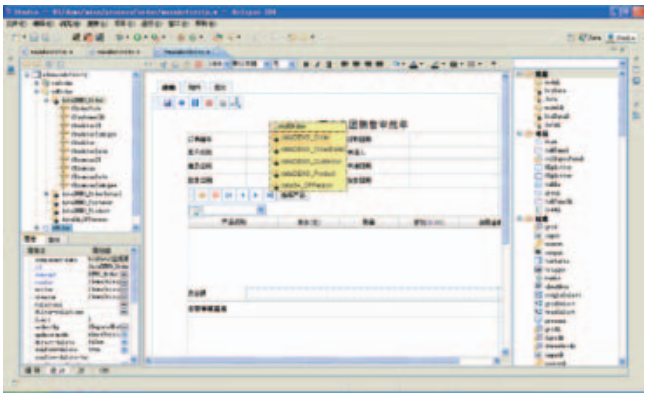
流程建模包括画流程图、设置流程环节的规则属性（比如执行规则、流转规则、回退规则、转发规则、通知规则等）、通过流程事件做流程相关Java扩展代码开发。

组织建模

组织模型包括建立组织机构、人员和业务权限。

界面建模

界面完全基于可视化的组件拖拽开发，设置组件的属性，通过界面事件做JavaScript扩展代码开发。表单布局既可以自由拖拽，也内置了Excel表格做布局，界面制作是很方便直观的。



如上图，界面中的要素都是组件，左下角的是组件的属性和事件，右边是组件面板，X5提供了很多的组件，当然也可第三方自己写组件。

通过上述四个建模步骤和相应的扩展代码开发，就可以完成信息化应用的开发，得到相应的应用系统。

业务建模的优势

业务建模的开发方式可以从许多方面大幅度地提升现有软件开发的层次。可以总结为如下几个方面。

产品质量：业务建模提供了从信息系统设计、开发到维护的全面质量保证

开发成本：业务建模能从多个方面极大地降低项目总体成本

开发周期：业务建模极大地加快开发进度

项目管理：业务建模大幅简化了软件工程，降低管理的复杂度

项目风险：支持快速应用，大大降低开发风险

人员要求：业务建模对人员要求大大降低

投资保护：使持续完善成为可能，大大延长项目和产品的使用寿命

关于Justep X5的更多详情和产品试用，
请访问www.justep.com

《小鸡快跑》的成功故事

文 / 刘江

一只来自中国的“小鸡”，在亿万应用的竞赛中，跑在前列。笔者深入发掘，为你带来“小鸡”成功背后的精彩故事。



来自中国的“小鸡”在世界上跑出了好名次

我习惯隔几天就在iPad上浏览美国区App Store排行榜，看看有没有什么新应用出现。7月份的一天，一款叫《Run, Run, Chicken》的游戏冲到了第二。下载试玩了一下，形式非常简单，但操作流畅，玩起来很有意思，非常适合打发碎片时间、换脑子放松用。游戏的图标风格和美工设计看上去似曾相识，很中国，我突然想到，这可能是一款国内团队开发的游戏！搜索之下，果然，游戏支持网站的联系方式有新浪微博。然后又发现，这款游戏在国内已经火了，中文译名叫“小鸡快跑”（由于和著名卡通片重名，SEO很不利，呵呵）。

经历传奇的负责人

在微博上，游戏的负责人 “@小小米卡”非常活跃，在不断报告小鸡的下载量超过N百万、对用户意见的反馈鞠躬尽瘁的同时，还顺带泄漏出自己的多才多艺：简历里写着“合格医生”，会英法日韩等八种语言，有画廊且画工颇精、每天都用画记日志，有工作室爱玩机器人捣腾Processing，在团队里可以做设计，也偶尔客串编程……当然，我也发现他和我的作息时间类似，都是深度夜猫子。这是一个怎样的团队呢？一天深夜通过微博，我约了前去拜访。

见了面才知道，小鸡快跑开发团队纤炫媒体科技（Xmedia）原来是3G门户的一部分，李幸福（小小米卡）也同时担任3G门户技术副总裁。李幸福经历非常传奇，出生在韩国首尔，童年在新加坡度过，后来在英国求学和工作。剑桥学医出身，曾供职于高盛和淡马锡。在IT行业有多年从业经验，与人合作针对ARM图形引擎ePage研发的eFIF，获得了13项专利。上次IT业萧条的时候，他在英国招揽了不少优秀人才，投资成立Mika Mobile，现在已经成为实力非常强的移动游戏团队之一，推出过App Store Top10游戏Battleheart。

2008年，李幸福又与清华毕业、在英国获得图像处理博士学位的老友陶冶刚到中国创业，研发了跨平台的图形技术Akebono（日语，意为曙光），并基于这一技术，在功能手机上实现了界面效果华丽的真3D UI浏览器OLA。2009年以Go浏览器的名称推向市场后，用户累计超过2500万，稳居移动浏览器前三名。

痛苦的转型

2010年，李幸福看到iPhone和Android为代表的智能手机兴起这一大趋势，开始思考转型，面向新时代从零开始。转型过程相当痛苦。整个团队在广州两个多月，摸索了各种思路。

- 做手机？要同时解决软件和硬件的诸多问题，很可能需要长期艰苦作战，前途未卜。
- 做UI解决方案？那就必须和手机厂商合作，OEM、授权等等，那是做大客户业务，风险太大，团队也很难做大。之前Mika在英国有公司产品曾经达到装机2亿多最后却功亏一篑的前车之鉴，当然也不愿意重蹈覆辙。
- 继续围绕现有的3D技术方案想出路？在中国光靠技术很难挣钱。

这时候，在英国的Mika Mobile传来好消息，随着几款iOS游戏上榜，公司的收入暴涨。这使李幸福下决心向智能手机平台、向游戏转型。当时已近年关，他没有马上向团队宣布这个重大决定，而是让大家安心过了一个春节。

2011年3月开始，除了继续开发和维护3G门户的Go浏览器和Go天气产品外，团队开始全面学着做游戏。“头三个月我们只是累积经验，培养信心”。李幸福说，“什么应用都可以尝试。”但并非每个人都有信心，很快有些同事离职，包括一个创始人和主美术设计师。所幸公司一位培养多日的小伙子顶了上来，李幸福对此很欣慰，因为这个年轻人原来在餐馆里打工……

这真是摸着石头过河。团队成员基本上都没有什么经验，就多方尝试，包括模仿Talking Cat。第一款成型的产品是模仿一些跳舞类游戏做了《Dance, Dance, Babe》。有意思的是，产品出来以后，大家发现这不像游戏啊，怎么玩啊，虽然3D形象舞步很逼真，也很流畅，技术上难题都解决了，可就是不像游戏。李幸福笑着回忆：“那时候我们都不知道怎么把它变成游戏。配上音乐以后，好像只是一款有3D形象伴舞的播放器……”是否将它推出？团队内部出现了分歧。最后是李幸福决定就这样发布1.0


游戏负责人李幸福有着传奇的经历

版本，看看市场的反映。结果，4月在App Store上线后，第一个月就冲到了中国区免费游戏第一，当月下载50万。

因为很多用户将《Dance, Dance, Babe》当游戏下载后发现没法玩，无法互动，最后的成绩也不过150多万。但这款产品给了团队极大的信心——原来并没有想象的那么难。另外一款几乎同时推出的Talking Bunnie也得到了Apple的推荐。4月15日，李幸福发出一条微博：

“随着《Talking Bunnie》和《Dance, Dance, Babe》的小试牛刀……我们的第三和第四个iPhone应用都是游戏……就快了……越来越有感觉了。”

小鸡火了

他说的第三个游戏，就是《Run,Run, Chicken》。

《Run, Run, Chicken》的最初创意来自团队的技术负责人陶冶刚。他本身是个游戏迷，原来开发浏览器时，旁边都会放一个PSP，敲敲代码完了拿个PSP玩一玩。当时他玩了几款奔跑类的游戏，就是一个动画形象一直跑、跳、走，速度非常快，又有很多门槛，很紧张，难度很大。其中包括《Line Runner》（由SkyOS的作者Robert Szeleney开发），听某人说已经有100万下载。他就想，这种游戏我们团队开发出来





李幸福认为美工是小鸡快跑成功的关键

没问题。

选用什么形象呢？一定要很可爱。从《愤怒的小鸟》想到用鸟类，又联想到著名卡通片《小鸡快跑》，小鸡形象就这样确定下来。

核心研发团队由5个人组成：两个美工，分别负责3D和2D；其他三个开发人员分别负责引擎、关卡和服务器。

《Run, Run, Chicken》最开始是收费游戏，定价0.99美元，可是三天只卖了100多美元。李幸福后来回忆，iOS里强手如云，竞争太激烈了，做得精致的好游戏不少，用户不可能在应用商店里翻很多页去主动找到你。

这时候，团队做出了一个至关重要的决策——免费，把眼光放长远一些，先打品牌，不考虑赢利。突然间，下载量暴涨了，从一天3~4万的下载量开始，然后连续两周平均都有10万，最高到了17~18万一天，而且都是在没有付费推广、没有打广告的情况下实现的。除了在美国区最好成绩达到总榜第2外，在139个地区商店登顶，217个地区进入过Top 5。上线40天，突破了500万，小鸡火了！

“小鸡”为什么成功

李幸福也承认，《小鸡快跑》的成功，幸运成分居多。“不知道为什么就火了。”谈及经验，他觉得最重要的是美工。在研发过程中，小鸡形象力求可爱、能第一眼抓住用户。具体设计了很多稿，精益求精，甚至到了快要发布之前，还改了一个版本，整个3D建模都要重新建，负责的人员几乎崩溃了。此外，3D场景也做了很多版本。

当然，适时采取了免费策略也很关键。陶冶刚是资深游戏玩家，所以开发出来的第一个版本特别难，没几个人能玩过第一关。李幸福建议将门槛降到最低。所谓五分钟娱乐，要尽量做简单的，操作简单，规则也不要太难太复杂，短时间玩一玩，解解压就行。在我看来，这个决策奠定了《小鸡快跑》风靡的基础。这种真正的玩家看上去挺弱智的游戏，恰恰适合许多和我类似之前基本上不怎么玩游戏的人。

在关卡设计、难度设计上，团队也花了很多心思。这个游戏的玩法简单，就是跳。但怎样调整难度，使前三个Level都能让用户感觉自己很牛，很快能通过，然后接下来又慢慢加道具，加难度，让用户觉得好玩，又有一定挑战。这个方案也改了好几次。

技术方面也解决了不少难题，因为游戏是3D的，维护增速有一些难度，性能上做了很多优化。

推广上，会八种语言的李幸福承担了大部分工作。因为主要面对国外用户，他经常需要半夜两三点、四五点还泡在Facebook、Twitter和Toucharcade等十几个论坛里。Toucharcade里一个游戏就一个帖子不停地回复，累计上万。他戏称自己不像做游戏的，倒像是炒外国股票

的。因为Mika Mobile的推广和客户反馈也是他在做，所以有用户基础，可以提前透露情况，引起用户的注意。在这一点上，小鸡快跑沾了兄弟公司的光。但产品口碑还是最重要的，《Dance, Dance, Babe》到了100多万关注的人就不太多了。对于国内用户的推广，新浪微博的效果不错。

此外，游戏的独立博客也发挥了不小的作用。定期将研发过程的一些亮点、一些决策公布，可以及时收取用户的反馈，这些反馈中有不少无形中变成了功能。其实也促使开发模式在改变，从原来预先拟定路线图一步步实现，到现在整个过程接近透明化，社会化程度很高。

有意思的是，这种将推广与研发结合起来的开放模式，是我之前设想过的，但没有料到实际中已经有团队实施。我提出了一个常见疑问："你们怕不怕透明化的时候，对手会去抄？"李幸福回答，即使你在研发过程中不开放，产品出来以后还是会有人抄。这个主要还是看实力，他们自己本来也是模仿别人，如果有人真的做得比他们成功，也只能服气。事实上，《小鸡快跑》之后很快就有《小猪快跑》，但是并不成功。相比之下，研发中与用户的交流更加重要，否则连成功都不可能，就更不用说怕被人抄了。

此外，他还表示，此前在做浏览器的时候已经发现，集中式的决策弊端很多。以前团队很多事情还是由他和陶冶刚主导，比较关注自我的想法，觉得产品越炫越好，哪个功能比较炫就做进去，不太注重用户反馈，结果做Demo很成功，做产品却是失败的。后来交棒给年轻人，放手让团队去发挥，反而取得了很好的效果。他们的经验就是用户主导。因为浏览器的用户量不断增大，反馈也非常多，仔细研究反馈，指导研发，整个开发模式与他们之前在国外已经有很大不同。

我问起研发过程中的主要教训有哪些？李幸福总结，如果最开始决策更快一些就好了。开始团队各个成员在一起开会，众说纷纭，每个人都有自己想做的游戏，想法非常多。也有很多原创的想法。如果当时继续那样发散的话，可能现在一个游戏还没做出来。速度太慢对小团队是致命的。年轻团队最重要的是积累经验，不妨从模仿入手。经验多了，自然会有原创的东西出来。

对未来的展望

对于未来，李幸福充满信心，他庆幸自己转型及时。在智能手机应用商店模式下，《小鸡快跑》的下载量已经比之前在功能手机上以SP模式做浏览器，通过3G门户的流量推广效果还要好，是最直接的证明。

当然，《小鸡快跑》虽然在下载量和品牌上都取得了成功，但游戏是免费的，没有什么广告，也没有应用内收费。未来团队如何赢利？李幸福认为，《小鸡快跑》可以利用3G门户的广告资源做一些植入性的广告，应该能取得较好的效果。但是，一个团队要发展，需要更大的规划，一年推出多少种产品，其中有明星主打产品，一些是免费、积累用户基数的，一些是收费的，一些是应用内付费销售道具的，需要同时保证数量和质量。目前这个团队仍然非常年轻，还在积累经验阶段，不着急。

今年，团队的目标是发布12款游戏，包括一款规模较大的RPG类游戏。产品的原则是不做区域化只针对中国或者亚洲的产品，一定是定位全球。比如，不会做三国类、比较卡哇伊的。

在操作系统方面，他认为先做好iOS的产品，建立口碑效应，转到其他平台相对比较容易。

除了iOS外，李幸福还看好Android，尤其是三星可能会成为一个主导性的厂商。他不看好Windows Phone这个平台，因为虽然微软很有实力，但在移动互联网上落后太多。而且作为小开发团队，得到的支持也很少。他相反对PSP等游戏机平台更感兴趣，事实上Mika Mobile已经有团队在针对这些游戏机开发。其他平台基本不再考虑，没有时间耗，需要集中精力聚焦在最强的平台。

做浏览器出身的他们对HTML5当然也非常关注，但显然短期内性能无法支撑游戏开发。现在不会在这上面花太多精力。



Scott Jenson
青蛙设计创意总监。20世纪80年代后期是苹果UI小组的第一批成员，从事System 7、Apple Human Interface指南以及Newton方面工作。之后，Scott Jenson担任了多年的自由设计顾问，然后成为Symbian系统的产品设计总监，并最终在Google管理移动UX组的工作。
本文获得www.frogdesign.com转载授权，如需引用请与frog design联系。

移动应用必定不会长久

文 / Scott Jenson

青蛙设计创意总监Scott Jenson认为移动应用虽然暂时繁华，但是“即时”交互的整体概念意味着这种景象并不会长久。

移动应用的历史是一个漫长而痛苦的过程，一开始是简单复制台式机的做法，然后窘迫地认识到，这种方式不太可行。其实，这是一切事物进步的方式，不仅在技术领域，艺术和音乐也遵循类似的模式，复制、延伸，最后发现一个新的模式。要摆脱旧的范式，需要耗费一段时间。

移动应用显然是成功的，并且在某些情况下，其盈利非常可观。如果我说它们必然消亡，这听上去明显与大量证据相违背。但所有的事物都会更迭交替，在快节奏的技术领域中尤为明显。

当一个范式发生转变时，很少是因为人们讨厌旧的模式或认为它没有用，而是因为它无法利用新的机遇。守旧派坚持自己的主张，愤怒地喊道：一切都完美无缺，你是在夸大其辞！

策划、组织和筛选过程将变得十分复杂

应用程序（这里指必须下载到手机上的本机应用程序）的问题在于，它们的组织和维护过程非常麻烦。让你要去的每一家商店、拥有的每一件产品和访问的每一个网站都拥有对应的应用，是不现实的。因为这将不断增加新的内容，必须对其加以策划、组织和筛选。

每个人都会执行一项常见任务，那就是隔几个月删除一次旧的和未使用的应用程序，有效地清除电话中的垃圾。条理性非常强的人会迅速清理这些，以避免手机上的应用程序数量增长过快，但大部分人对此并不在意，因此他们的手机主页将陷入到不断隐没的复杂选项中。

就其本身而言，这个问题显然不会为应用程序带来致命影响，但它确实显示出一个不断扩大的问题。如果我们想要的应用程序数量是现在的两倍，甚至五十倍，将会怎样？我们还能坚持有效地进行策划、组织和筛选吗？

这会让人想起早期的Web，当时Yahoo!有一个固定的网站层次结构，但随着Web的爆炸式发展，该结构变得越来越难以维持。Google打破了这种既定范式，通过消除所有组织结构，只使用搜索系统，从根本上避免了这个问题，用户因此得以方便快捷地访问数以百万计的页面。

用户体验的金科玉律：价值大于使用难度

有一个微妙的力量在起作用。人们在讨论任何产品、技术或想法时，很容易只注重它的价值，即它试图为用户解决什么问题。这是一个良好的开端，并且历来都是人们唯一考虑的因素。然而，近来人们已经开始认识到，产品必须经过精心的设计，使其易于使用。

任何产品的价值及其用户体验，通常被作为独立变量来对待。你当然希望得到一个高价值的产品，与此同样重要的是其使用的困难程度也必须较低。尽管人们不愿意承认，但这两方面是紧密联系的。最重要的是

它们之间的关系：价值必须大于使用难度。

例如，早期的SMS系统极其难用，但其价值（避免了昂贵的按分钟计费）很高，超越了使用难度。当然，SMS体验的改进大大增加了使用率并吸引了更多用户。仅仅是价值大于使用难度并不意味着大功告成，只说明它质量高，足以向市场推出。

但这一模式还解释了另一个更重要的因素，即用户行为。随着使用难度下降，人们将更频繁地使用产品来完成价值较低的任务。价值仍然大于使用难度，但现在触发使用所需要的价值比以前低得多。最典型的例子莫过于进行Google搜索。Google曾公开表示，如果把Google主页的加载速度减少零点几秒，其使用率就会明显提高。Google主页没有以任何用户认知的方式进行改变，它只是快了一点点，就会有更多人使用它。这很重要，因为只是降低使用难度，而不以任何方式提高产品的价值，会显著影响使用率。

因此，回到本机应用程序的使用。如果你要走进一家商店，店员自豪地在门口宣布他们有一个应用程序，你会立即安装它吗？你会在头脑中进行怎样的价值/使用难度计算？如果你是该品牌的粉丝，那么对其价值的认知可能较高，可能愿意忍受繁杂的安装过程，这时价值大于使用难度。不过，如果你从来没有听说过这家商店，你可能不会在意，这时价值小于使用难度，那你就不愿意冒险尝试。

我想说的是，如果应用程序以某种方式奇迹般地出现在你的手机上，你会更有可能尝试它吗？当然会，因为免去了复杂过程，使用难度为零。你会尝试这个应用程序，因为你不会有任何损失。我们要想出一个办法，消除使用应用程序的难度。

本机应用程序，使用难度远大于零

让用户负责应用程序管理，实际上给他们带来了稳步增加的使用难度。这使得应用程序的压力越来越大，它们的使用频率会越来越低。这不是一个绝对的问题，而是Google主页使用率的一个反面例子。随着事务的复杂性和烦琐程度缓慢增加，使用率将开始一点点下降，刚开始难以衡量，但随着时间的推移可能会越来越明显。

如果你还是无动于衷，我仍然可以理解。你可能在想：“说真的，你需要多少应用程序？问题不会变得那么糟糕，对吧？”在技术领域的人必须始终牢记，没有什么是稳定的。今天似乎完全合理的，但在很短时间内就会变成束缚。我敢肯定你们大部分人都记得，原来的PC DOS团队预期不可能有人会使用超过640KB的RAM。

它可能会变得更糟糕……

可能会彻底改变我们目前微小的应用程序世界的是直线下降的计算和连接成本。这将大大增加我们每天接触的设备种类和数量。

事实上，很可能我们将经历数百甚至数千台设备，每台设备都将能够为我提供某种类型的交互体验。我的博客文章——《僵尸末日》（http://designmind.frogdesign.com/blog/the-coming-zombie-apocalypse-small-cheap-devices-will-disrupt-our-old-school-ux-assumptions.htm）中有更详细的描述，但智能设备的成本正在下降，在许多情况下，已接近于零。下面是一些例子。

> 移动应用程序必然消亡的原因是，它是一个阻挡我们前进的范式。“即时”交互的整体概念，从结构上来说无法利用安装的应用程序来实现。

- 如带有RFID或NFC射频标签的电影海报，能让我在手机上获得海报的交互式版本，为我提供更多信息。
- 任何消费项目，像番茄酱或奶瓶，也带有射频标签，它们将不仅像海报一样能让我获得关于这些项目的更多信息，而且还能跟踪使用情况，甚至提供替代品的购买。
- 我的本地巴士站将被地理定位，所以我目前仅需安装GPS，就可以获得这个特定的巴士站的信息，知道下一班车将在什么时候到达。虽然使用一些复杂的城市系统在今天可以实现这项功能，但地理定位系统的部署让任何一个城市可以在所有公交线路上更便宜地实现这项功能。
- 当我走过任何商店的门口时，都将有一个

应用程序可以与我交互。

- 无论我何时去逛大型购物商场，它都将提供地图和时间。
- 本地食品车的供应商提供的将不仅是它们的菜单，还有它们准备去的下一个目的地以及返回的时间。
- 如Zipcar等按需租凭的汽车租赁公司，将允许我使用手机上的蓝牙连接来注册并开走它们的一辆车。

“即时”交互

所有这些概念只是猜测，但它们代表了一种正朝我们迎面袭来的趋势。这些设备都可能需要某种形式的交互，但只是在我接近它们时才进行交互，这是一种“即时”的交互模式，只在我需要它时，为我提供交互性。

更重要的是，绝大多数交互将是一次性体验。我将与设备（像海报）进行几秒钟的交互，然后继续做其他事情。这是一种“先使用后失去”的情况。在这种情况下，下载到手机上与该设备进行交互的任何东西在交互完成后都不再需要了。

这就是为什么这种交互将最有可能以某种Web页面形式存在的原因。以不涉及安装的方式把交互体验下载到任何设备上，是一种简单的方法。对于“先使用后失去”方法，Web是适合的选择。

事实上，Web在20多年来一直只提供这种方式。关于那些声称Web不能作为本机应用程序的说法，目前事实的确如此。但请记住：首先，标准正在非常迅速地提高；其次，与这些小型廉价设备进行的交互是相当有限的，就像与一个公共汽车站进行交互时并不需要具有运行“魔兽世界”时的强大功能。

今天，我不能够或设想利用我们当前的应用程序模式来尝试这类的“即时”交互。搜索、下载、使用应用程序，以及最重要的删除一个应用程序，这其中所耗费的精力实在过于巨大。

移动应用程序必然消亡的原因是，它是一个阻挡我们前进的范式。“即时”交互的整体概念，从结构上来说无法利用安装的应用程序来实现。

探索服务

快速打开智能设备并与之交互是不可能的，除非能迅速地找到在你面前的那台设备。为了毫不费力地与一天中将会经手的集群设备进行交互，我将需要一个可以使用手机上蓝牙、NFC、GPS和 WiFi不断搜索功能的服务，不仅可以找到附近的设备，还可对其进行排名。这并不是大海捞针，而是在针堆里找针。可能会需要一些云服务器的帮助，因为它们对我的了解更多一点，可以对这些设备进行合理的排序。

我强烈地感觉到，这类探索服务将在未来几年内成为下一个Google。这是Google、苹果和微软都整装待发要立即解决的问题。甚至可以尝试执行一个巧妙的启动，由系统发现在我身边的物理设备并对其进行排名，现在这个做法几乎变成了必不可少的服务。

在我点击每个附近的设备之后，就会打开在该设备或者更可能是在中央服务器上提供的一个 Web页面。实际会发生的情况完全取决于那台设备，和我并没有关系。正如现在对本机应用程序的限制非常少，那么对这些智能设备的限制也应该很少。该系统的目的只是识别和提供“即时”的功能。

展望未来

本机应用程序是计算机历史侏罗纪时代的遗留物，是阻挡我们发展的局部最大限制。探索服务和“即时”交互的结合，是一种本机应用程序无法提供的功能强大的交互模式。

今天没有谁可以提出像这样的模式。事实上，实现这种模式很可能会是一个长期缓慢的过程。但如果继续尊崇本机孤岛功能，我们甚至都不知道这是否能实现。在进行构建之前，你必须知道你想要的是什么。

责任编辑：陈博（chenbo@csdn.net）

从数据角度看手机游戏产业中的佼佼者

文 / 郑金条

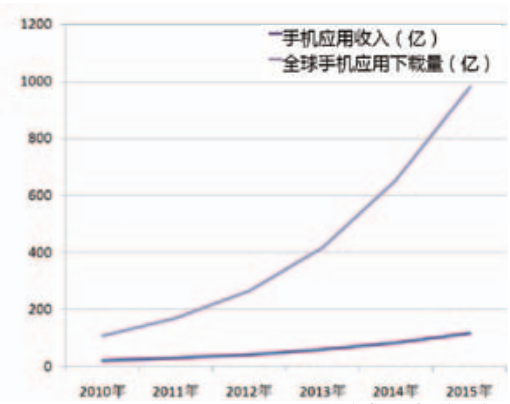
移动游戏的平台前景如何？营销推广趋势将有何改变？国内外市场还具备怎样的潜力？谁是目前行业的佼佼者？本文通过数据一一为你解答。

数据预测和行业发展趋势

Berg Insight的研究称，2015年全球手机应用下载将达到980亿次，从2010~2015年复合年增长率将为56.6% ，而营收也将从去年的21.5亿美元上升到2015年的118亿美元，其中亚太地区下载量占比将在2015年超过40%。

尽管数据光鲜，但并不意味着所有的开发者都能随随便便在这份创业热土上获得成功。来自Appsfire的数据显示，iOS和Android应用总共超过100万（其iOS应用占比52%，Android应用占比48%），大约80%处于活跃状态。换句话说，这同时也是一个超级拥堵的市场，开发者除了在游戏研发上需要费心权衡，同样如何让游戏在过分竞争的市场有一席曝光之地，也是让所有人犯愁的老问题。

质量变成了未来角逐手机游戏市场可能成功的一个关键要素，滥竽充数已经不能赢得市场，市场需要价值元素（The price of everything, the value of nothing）。当我们看到Distimo发布数据称Google Android Market有20%的免费应用



全球手机应用发展趋势



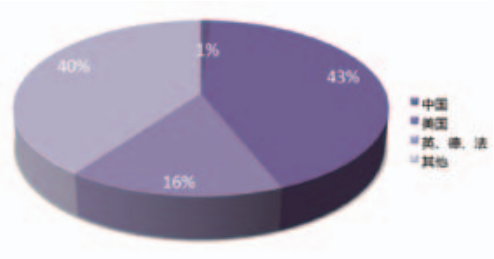
Streaming Colour开发者调查（基于225位）

和80%的付费应用下载量低于100次时，其实大家都有足够的理由恐慌，这是一个可以随时将心血打水漂的险恶淘金地带，即使开发者掏心掏肺好不容易设想了一个，也随时可以列举出可能遭遇的强劲竞争对手，比如Zynga的新游戏Dream Zoo在苹果App Store遭遇的阻击对手就有Pocket Gems旗下的Tap Zoo、TinyCo旗下的Tiny Zoo Friends、Beeline Interactive旗下的Monster Pet Shop、Backflip Studios旗下的DragonVale；即使开发者宣布游戏免费，试图从免费的IAP模式中获得一点利益分享，有时候也略微显得奢侈了。Snappy Touch称手机游戏免费模式的IAP营收80%~90%来自0.5%的用户，而这些用户在哪里往往还是一个可怕的未知数。

Streaming Colour关于iOS开发者营收情况的调查（采样对象是252名开发者）可能更显窘迫，在Lifetime Revenue这项数据上营收在1万美元以下的占比62.7%，在1万~10万美元之间的为21.8%，在10万~100万美元之间的为11.5% ，不能有效获得营收的则占了绝大部分的比例。1%的开发者获得了36%的整体营收，19%的开发者

获得了61%的整体营收，而更为广大的80%的开发者则仅能获得3%的部分。

关于国内外市场的分析



AppAnnie ： iPhone应用营收国家分布

我们每次发布欧美数据总是有人第一时间联想到中国市场的表现，其实在营收规模上，以iPhone应用营收为例，App Annie称中国占比1%（尽管中国的iPhone应用下载量仅次于美国，意味着其实有巨大的下载市场，但在越狱和其他非法下载上面的泛滥使得营收成为理论可能），而美国占比43%（iPad应用方面中国占比5%，而美国占比42%，欧洲主体市场的表现也不俗，英国、德国和法国三大欧洲国家在iPhone应用方面占比16%，iPad应用占比14%），对于期待能够从用户下载和游戏行为中获得更有效收益的开发者来讲，海外市场（主要集中在欧美以及华人市场）目前的吸引力远远高于本土市场（国内更多受制于政策因素、支付因素、盗版泛滥以及更可怕的用户免费心态），因为没有分级的问题部分游戏题材（黑帮或者血腥题材）并不能获得发行资格；另外一个问题是山寨，不劳而获和担心被山寨的双重矛盾心态让国内的创新基本停留在理论层面。在我们向山寨曾经的繁荣致意时，山寨将是我们行业发展路上的一杯毒酒，饮鸩止渴的暂时满足感最终将戕害行业的规范性。Copy To China（Yes）和Create In China（No）让人充满着对未来的担忧。不少的开发者认同先作恶再从良，这一点足以让创新偃旗息鼓。尽管海外市场并不都是灵丹妙药，包括作品的文化差异、营销渠道的差异甚至当地法律法规的差异也都同样致命，但相比较而言，海外的吸引力仍然是开发者发布游戏的首要选择。

同样在海外营销的选择上，一般也更加集中在苹果App Store和Android Market这两个超级渠道上。

一个最基本的数据预测，Xyologic认为到2013年中期，Android应用下载量将赶超iOS应用的总体下载量，这两个平台届时的应用总下载量都将达到840亿次左右，目前苹果App Store的应用总下载量为182亿次，而Android Maarket的应用总下载量则是67.6亿次。当然另外一个数据则对于开发者相对就不是那么友善，到2012年中期，Android平台的应用和苹果App Store两者届时的应用总数都将达到68万款左右（总共136万的应用，简直是天文数字）。尽管Berg Insight预测，到2015年全球手机应用下载将达到980亿次的说法也很诱人，但面对超过136万的竞争选择，相比未来应用曝光仍然是艰巨的问题。两个月前游戏邦曾经探讨过应用配套产业的存在价值，就认为应用维护、分析、销售和推广服务等环节将更加重要，Research2guidance认为这种重要性将有30%的产业价值， 2015年规模达300亿美元（到那个时候，赚开发者的钱同样也是巨大的环节，收费对象就不一定是直接的用户了）。



手机应用下载量分布副本

当然，至于苹果App Store和Android Market最终哪个对于开发者更有竞争力则只能见仁见智了，但至少在目前为止，苹果App Store在开发者群体中的号召力更胜一筹。IHS iSuppli认为2011年苹果App Store营收将达29.1亿美元，比去年飙升63.4%（去年营收17.82亿美元），全球占比76%，这个数据直截了当，与之对应的是2011年Android Market产值预测为4.25亿美元。Game Insight旗下的Paradise Island五月在苹果App Store和Android Market的收益比值可以更直观一些，Paradise Island在Android Market

的营收为64万美元，在苹果App Store的营收则为300万美元，这意味着两者的比值是1:5；GluMobile之前也提到过他们在Android和iOS两个平台的收益比值在1:8或者1:10之间，刚好个体和整体的比值都稍微能匹配。

当然在非大陆地区的市场考量上，也可以考虑下ARPU值很高的中国台湾市场的表现。根据中国台湾资策会的相关解析数据称2010年第二季度台湾手机上网用户为490万，其中159万下载过应用（白领和大中专学生为主要下载群体），市场份额依次是Android为35.2%，苹果iOS为24.6%，而Symbian和Windows Phone分别为13%和60%。而应用下载的绝大部分为免费下载（60%），但对于付费用户来讲，其ARPU值却显得相当不俗，付费下载用户每月人均花费305元新台币（约为10美元）。在中国台湾市场上，远传电讯旗下的S市集引入Gameloft游戏，而台湾大哥大也和EA合作引入8款Android手机游戏。

市场的出路和营销收益

在考虑收益产值以前，可以先关注下用户行为。在早先的一份数据中，iPhone用户家庭收入在10万美元以上的比例为47%，5万~10万美元之间的为34%，低于5万美元的为19%，这倒回应了我们前文提到的问题：为什么目前为止苹果App Store在游戏营收上会更有竞争力【注：Hunch.com八月发布的iOS和Android平台对比信息图显示，24%的Android用户家庭年均收入为5万~10万美元，67%用家庭年均收入达20万以上，链接为http://gamerboom.com/archives/35374】。

当然，考虑用户行为Flurry的数据就值得一再借鉴。在Flurry的数据比照中，游戏的免费概念（前100名中大概有65%为免费模式游戏，主要通过IAP营收）已经一再深入玩家的行为中，按照他们对数千万用户行为的监测中显示，男女用户玩免费游戏的时间比例为53:47；在免费游戏的消费金额中，男性和女性二者比例为58:42；在虚拟支付方面，男性和女性投入分别是15.6美元和11.9美元。

我们前文提到如何在无数的应用中脱颖而出是值得开发者思考的，另外一个同样需要思考的是，在获得用户以后开发者需要用什么样的方式让玩家持续付费。

先阐述第一个问题，在渠道方面，需要根据投放市场和用户定位进行重新权衡，一般可以分为线下渠道、手机渠道和PC渠道。线下渠道则包括了电视广告（比如Angry Birds在超级腕的广告）、电影广告（比如Lima Sky和Halfbrick和影院方面的合作）；手机渠道则包括了手机广告（比如Google的AdMob）、交叉推广（比如迪士尼游戏的交叉推广）、第三方推广（比如Tapjoy提供的服务或者Heyzap的展示推广）；PC渠道则包括了传统在线广告（通常为Banner形式）、专栏评论（比如Pocketgamer）、推荐网站（比如Free App A Day）、应用评级和用户评论（比如Quality Index）以及社交营销（比如Facebook、Twitter或者微博）。国内顶尖的厂商Haypi旗下的Haypi Kingdom则提供了一个直观的模式：渠道投放约15万美元，主要分布在Free App A Day（1.5万美元）、Tapjoy（约5万美元）、Google AdMob（约5万美元）、Apple iAd（约1万美元）。

而在第二个问题上，有一个数据可能更可以说明问题，在苹果App Store的应用营收中有72%（24%来自含IAP的付费应用，有48%来自含IAP的免费应用）来自IAP，而开通IAP功能的游戏仅占整体应用数量的4%。于是又绕到了刚才提到Flurry的分析，未来IAP将占据APP营收的主要格局，而徘徊在直接免费以IAP营收还是通过LITE版后吸引付费下载创收的开发者或者可以看看这个数据比照：通过IAP交易额比多数开发者通过精装版+付费版这种盈利模式的收益高14倍。尽管不是每个人都乐意参与IAP，但参与的用户平均消费额度相当可观为14美元，按照Flurry的说法，这一切仍然仍然符合二八法则，少部分的消费者（13%消费20美元以上）贡献了大部分收益（51%营收），71%消费力在10美元以下的，只贡献了整体营收的 31%。

那些典型的成功者

尽管数据分析相当悲观，但能够洞见行业发展趋势和用户需求而大获成功的开发者也并

不缺少，他们散落在全球各个区域，成为各自开发领域的佼佼者。

最典型的成功莫过于Epic Games旗下的



App Store收入方式分布

Infinity Blade，这款毁誉参半（很多人对生生不息的重复充满了厌倦，但同样有人对游戏的画面造诣充满了赞叹）的游戏售价高达5.99美元，并且还加入了IAP成分，却在短短不到一年的时候获得了超过2000万美元的营收（其投入成本仅有200万美元左右）。

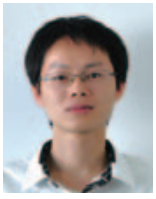
Storm8的表现也相当强劲，其在成功融资3亿美元后公司整体估值已经达到10亿美元的水准，Storm8首席执行官Perry Tam称Storm8（包括旗下的另外一家超级公司TeamLava）发布的每一款游戏都能进入到iOS免费榜的前十名（连续成功，目前在Android Market榜单上前十名就有三款来自TeamLava），每日活跃用户在400万以上，我记得他们在促销时曾有单日营收100万美元的记录。

第二季度三大上市公司营收表现	
DeNa	4.49亿（美元）
EA Mobile	5700万（美元）
Gameloft	3270万（美元）

第二季度三大上市公司营收表现

其他一些上市公司上季度发布的财报也能够显示出不俗的营收表现，DeNA一马当先季度营收达到了4.49亿美元，EA Mobile也有5700万美元的收益，Gameloft则有3270万美元，Konami在2012财年第一季社交/手机游戏营收为9900万美元（Dragon Collection一直是Gree上的头牌游戏），Glu Mobile（940万美元）以及Gamevil（560万美元）也各自有漂亮的财报数据。再远一点Beeline Interactive得益于Monster Pet Shop和Smurfs' Village的强劲表现今年第一季度营收超过1550万美元。而一些没有上市的超级公司（要么拥有超级下载量，要么有数款游戏能够进入营收前50名，比如SNL Kagan版的10大手机游戏发行商EA Mobile、 Gameloft、ngmoco 、Rovio Mobile、Glu Mobile、Capcom、Zynga、Sony Pictures Mobile、Namco Bandai以及MocoSpace或者Pocketgamer版的十大手机发行商Gameloft、Rovio、 Firemint、Chair Entertainment、PopCap、ngmoco、Halfbrick、Capcom Mobile、Backflip Studios以及EA Mobile）则只能根据旗下游戏表现的好坏来揣测可能性的营收结果，比如Pocket Gems上个财季被预测整体营收可能超过1500万美元，SlideShare则有PPT搜集了3月份大量游戏开发者的单月营收数据，尽管没有经过官方证实但也可以权作参考（Ngmoco，284万美元；Capcom，278万美元；Pocket Gems，274万美元；Storm8，161万美元；Addmired，104万美元；The Playforge，100万美元（力作是僵尸农场）；Team Lava，94万美元；Gameview，62万美元；Pokerist，60万美元）。

至于在单款游戏上，就如我们前文提到的Infinity Blade有超过2000万美元的收益；Happylatte 旗下的High Noon据说在7月单月也有超过100万美元营收的表现（下载超过600万次）；据说Haypi旗下的Haypi Kingdom游戏海外营收已经超过500万美元；Pocket Gems旗下的Tap Zoo也被认为月营收高达数百万美元，而时下火爆的游戏Tiny Tower也有人根据其MAU、DAU、DAU/MAU比值、IAP转化率和玩家收购虚拟商品的种类（0.99美元占45%，4.99美元占51%，29.99美元占4%）等进行加成推测Tiny Tower的月营收可能在30万美元以上。



郑金条

游戏邦负责人，游戏邦主要关注和解析国内外社交游戏和手机游戏领域，并定期做深度行业阐述。

责任编辑：陈博（chenbo@csdn.net）

“建立企业中的用户体验力量”系列之三

用户体验的工作方式

文 / 吴卓浩

本文带你从用户出发，到用户检验；从充分探索，到迭代改进。以正确的方式思考，有效参与用户体验工作。

用户体验是用户在使用产品、系统、服务的过程中产生的感受。用户体验基于个人感受，但在确定的用户群以及使用情景下，反映为群体的共性。这是用户体验研究与设计的基础，用户体验的工作也是围绕着这个基础展开的：由少量用户入手，了解和洞察他们现实和潜在的需求，探索满足这些需求的方法；然后经过大量用户的检验，逐步改进、形成解决方案。这些用户体验的工作是如何开展的呢？

从少量用户开始



用户体验工作是个从小规模到大规模探索、验证的过程

用户体验基于个人感受，但个人感受并不一定和群体共性相吻合。简单地说，每个人的看法都不相同，那么究竟应该按照谁的意见做呢？我给出的答案是：按照能够代表群体共性的意见。用户体验设计的基础和目的，就是找到用户群在获得体验以及产生反馈上的共性，从而通过设计来引导特定体验的发生。

一方面，人与人的偏好各不相同，但这些偏好背后的基础需求往往具有共性，而且越基础的需求越容易产生共性。比如在产品中究竟应该用红色还是黄色表示强调信息，部分开发者会为此争执不休。然而这个问题本身应该是如何表现强调信息，那么除了颜色，文字大小、字体和风格、出现位置、动态效果等都是可选的方式，并且有的方式可能比用颜色更好。

另一方面，当进行用户研究时，随着用户数量的增加，发现的群体共性会慢慢显现，同时，可供发现的新共性也会逐步减少。在实际项目中，针对某个基础问题对目标用户进行研究，往往经过十几位用户以后，获得新发现的概率就大大降低，这意味着大部分的群体共性已经被挖掘。由此就可以集中力量发展出少数几个解决方案，然后让产品上线，进行大量用户的测试和数据分析。

另外，从少数用户开始，本身也是对开发者从自身开始做产品设计的一种修正——无论你是否真的对产品具有敏锐的直觉，找一些目标用户来验证自己的想法，总归是个经济有效的方法。当然，因为是少量用户，他们的反馈也不能直接作为代表群体共性的结论，而是应该作为引发思考的线索，逐渐去逼近满足群体共性的解决方案。

了解和洞察用户现实的和潜在的需求

用户所想的、所说的、所做的往往不是一回事。

一种情况是，你问一个用户用不用某类产品，他说用，然后给你介绍这类产品的特点、行业趋势等。然后你问，你常用其中的哪些产品，他说出几个名字。你再问，那能不能演示一下怎么使用某具体产品，他却手足无措了，因为可能他其实没怎么用过这个产品。所以说不能只听用户怎么说，一定要看他怎么做。

另一种情况是，在与用户交谈或者观察用户操作时，有意无意地问他是不是要进行某种

操作，甚至替他说出答案或者动手帮他完成操作，这些在用户研究上也是一定要避免的，因为干扰了用户的正常自主操作，甚至在用户求助时，也不能给出有倾向性的建议。你需要问开放性的问题，保持观察者的立场，避免自己的言行对用户的判断和行为产生影响。

还有一种情况，普通用户通常意识不到或者无法正确表述自己的需求。而获得洞察的途径，可以是基于对主流用户或者极端用户的深入观察，可以是基于对用户数据的分析，也可以是基于自身作为超级用户的直觉。相比之下，基于观察主流用户或者分析主流用户数据的方法似乎更容易被人接受；但主流用户的行为以及数据更偏向于反映现状，超级用户或者极端用户更偏向于预示未来。所以基于主流用户的研究往往带来的是渐进的小改动，而基于超级用户或者极端用户的研究则更能带来突破性的进展。比如在对一个产品仅进行用户测试时，通常需要三种用户来进行测试：新手用户、主流用户、高级用户。新手用户对这类产品不熟悉，甚至对于使用设备、互联网都不太熟悉，这样的用户在试用产品时能更透彻地反映出产品在可用性上的缺陷；高级用户一方面会用一些高级的使用方法挑战产品的性能，另一方面他们往往是产品内容的主要贡献者和导向者，帮助他们更高效地使用产品实际上就是让他们更高效地为产品做贡献；主流用户在用户数量上来说是产品的中坚力量，产品对于新手用户和高级用户的优化不应该影响到这大部分用户的使用。

> 虽说历史上所有产品都是要经过大量用户检验的，但是互联网、移动互联网产品与以往产品最大的不同在于，可以迅速获得大量真实的用户数据，检验的结果可以迅速转化为产品的改进。

探索满足用户需求的方法

这个环节需要各种不同职能的人员通力协作，但同时这也是最能发挥用户体验设计师长处的地方：设计师基本不用考虑编程实现、不太需要考虑项目协调，于是把全部精力放在了解用户、探索满足用户需求的方法上。设计师所接受的主要专业训练：一是设计思维，即创造性、系统性的发现解决问题的方法；二是设计表达，即快速有效地把想法变为看得见、摸得着的设计效果图或者模型。

在很多企业中，设计师的职能被定位为执行者，当产品经理制订好产品设计之后，交由设计师完成具体的界面设计，并配合工程师实施执行。但事实上，用户体验设计对于整个团队的最大贡献是充分高效地探索问题本身及其解决方案。设计师无法让工程师编程更快，无法让开发所需的代码量变少，但用户体验人员可以通过准确把握用户需求、充分探索解决方案、以快速原型检验解决方案，更有效地规避风险、找到正确的路。从全局上优化研发过程和效果，这其实是比做好执行更重要的贡献。

在探索解决方案的过程中，特别注意不能急着提出解决方案，而需要从深入思考用户的根本需求入手。面对用户需求，开发者迅速做出一个相应的解决方案。这种快速反应是对的，但经常见到的情况是，由于没有真正分析和思考这个表面需求的背后原因，变成头痛医头、脚痛医脚，产品做不深入，而且总会产生新问题。正如前面所说的，用户通常意识不到或者无法正确表述自己的需求，表象不代表真实的需求；用户会表达自己的需求，但通常不会考虑自己不同需求之间的关系，当然也不会从产品 全局的角度来考虑满足不同需求之间的平衡；用户的表达往往带有情感，而这种情感的强弱会掩盖问题本身的轻重缓急。开发者必须要全面深入地思考用户的需求究竟是什么，很多开发者充满激情、执行力强，见到问题就想马上解决，这是一种优秀的素质。但如果没有在动手前充分思考，这就变成了一种鲁莽的行为，就会付出代价。

找寻问题解决方案的过程是一个不断发散和收敛的过程。你可以迅速提出一个解决方案，可是，你怎么确定这是最好的方案？最终的结果可能是一个方案，这当然很好；也可能是几个方案，无法直接判定哪个最好，那么可以上线通过大量用户的检验来做出选择。经过

尝试不同方案而得到的最后结果，和刚见到问题时脑子里冒出来的“最好”的方案相比，哪个更好？虽然你可以通过能力和经验的积累，让自己的直觉更准、更迅速地找到最优解，但寻求最优解的过程就是这样不断发散和收敛的过程，那种凭直觉一击命中的情况无法成为一种有效操作的方法。

另外，充分探索解决方案不是不计成本、无止境地尝试不同的解决方案而不做最终决定，迅速有效地找到接近最优的解决方案永远是最重要的。

通过大量用户检验

前面所做的所有工作都是为了接下来这一步，让产品经过大量用户的检验，才能知道是否成功。虽说历史上所有产品都是要经过大量用户检验的，但是互联网、移动互联网产品与以往产品最大的不同在于，由于可以迅速获得大量真实的用户数据，检验的结果可以迅速转化为产品的改进。过去的产品如果用户用的不爽，就会被遗弃；互联网、移动互联网产品则往往在用户用的不爽、还没有被遗弃之前就改进了，于是用户开心的继续使用。所以互联网、移动互联网产品的设计原则之一就是快速上线、快速改进。

但是这个“快速”原则有一个大前提，就是前面的工作还是要首先做好，由少量用户入手，了解和洞察他们现实的和潜在的需求，探索满足这些需求的方法，然后才是大量用户检验。如果没有做好这个前提，要么最终选取的方案离最优解相差太远，要么面对太多方案不知孰优孰劣、也无法进行大量用户检验。虽然快速上线、快速改进本质上是一种试错法，但无论这种方法在互联网、移动互联网领域多么有效，只有把前面的工作做好，才能真正产生有价值的方案可供检验。

在快速上线、快速改进的快速迭代过程中，执行力空前重要。高效地提供高质量的设计并实施当然是一个基本的要求，但是在这个过程中要想打得准、改得快，并不是只靠提高战斗力和士气就能做得好的，也可以从工作方



探索解决方案是个不断发散和收敛的过程

法和流程上入手。一方面要高效地探索解决方案，尽可能优选方案，从全局上提高效率和效果。另一方面是在设计结构和实现结构上预先考虑到修改的灵活性，不要一有修改就导致大量的工作推倒重来。最可怕的还不是有修改导致返工，而是因为畏惧返工而有意无意地回避尝试和改进。另一个是易于改进、多改进的良性循环，一个是不易改进、少改进的恶性循环，二者的效果不言而喻。

这些用户体验工作并不是用户体验人员单独完成的，需要整个大团队、不同职能的人员一起通力协作。虽说用户体验人员接受过专业训练和积累，在思维、表达、解决问题的方式上有独特优势，但是只有大家都为了最终的用户体验思考，才能事半功倍。用户体验是一门科学，是基于个体主观感受的群体共性选择的结果，并不是说每个人的个体感受就等于“用户体验”。找到和引导用户的群体共性是用户体验工作的重要内容。

从用户出发，到用户检验；从充分探索，到迭代改进。虽然用户体验人员是经过专业训练的专业人才，但用户体验工作首先是一种思维方式，而非一种特殊技能。只要以正确的方式思考，就可以有效参与用户体验工作。



吴卓浩

创新工场用户体验总监，曾创建Google在中国的用户体验团队。本系列文章将分享其在不同环境中建设用户体验团队的经验，讨论如何在企业中建立用户体验力量。

责任编辑：陈博（chenbo@csdn.net）

三星 bada2.0平台特性

文 / 李欣

三星公司最近发布了bada2.0版本，让我们看一下bada2.0有哪些亮点。

多任务

bada2.0支持多任务。只有一个bada应用程序可以在前台运行，其他的应用程序必须在后台运行。用户可以切换应用程序，决定在任何特定时刻哪个应用程序运行在前台。每个激活应用程序的大小也影响着多任务的执行。

bada2.0用户界面

bada2.0引入三种新的导航风格控件：Header，Footer和TabBar。

Header控件充当占位符的作用，在屏幕顶端，显示描述性的内容；Footer控件显示在屏幕底部，用于不同视图之间的切换或者放置按钮完成用户动作，TabBar是一个水平菜单，可以放置在屏幕主体区域的任何位置。TabBar控件可以滚动，并且包含多个项目，可能包含超过可以同时容纳到屏幕的条目数。

更多控件，比如SearchBar和Gallery也引入到bada2.0，方便UI功能的实现。从bada2.0版本开始，可以使用新的Osp::Ui::Animations命名空间。Animations命名空间提供的类可以帮助您的应用程序引入各种梦幻般的动画。Animations可以附加到Form或者控件上，实现各种效果。

布局管理

三星bada2.0支持三种不同的屏幕分辨率：WVGA [480x800]、 HVGA [320x480] 和 WQVGA [240x400]。可以使用三种不同的布局类型（线性布局、网格布局和相对布局）定位屏幕上的各种控件。

- 线性布局：UI控件相对于前一个控件被添加。有两种风格：横向和纵向。横向风格，新控件相对于前一个控件被添加到左边或者右边。而纵向风格，新控件相对于前一个控件被添加到上边或者下边。
- 网格布局：每个控件被放置在二维网格的单元格中。网格每个单元格的尺寸取决于放置其内控件的高度和宽度。
- 相对布局：每个控件的尺寸和位置是容器上其他控件的函数。

支持近场通信

近场通信服务是一种短距离无线技术服务。它包括非接触智能卡技术、RFID技术和近场通信技术。它使设备共享小于4厘米距离内最大通信速度高达424kbps的信息。NFC服务有三种操作模式：卡仿真模式、读/写模式、点对点模式。NFC设备可以改变操作模式。目前，bada NFC API只支持读/写模式。

增强应用程序间通信

为了支持多个bada应用程序之间的交互，我们引入两个类：ServerChannel和ClientChannel。一个应用程序可以使用ClientChannel向另外一个应用程序发送请求。请求由接收应用程序的ServerChannel处理，它可以控制被传递的响应。

支持语音到文本，文本到语音

bada2.0版本支持语音到文本类，可以识别语音并转换成文本。它也支持各种特性，例如语言环境、语法、静音监测、脏话过滤、标点符号忽略。

同时，文本到语音类合成一段给定文本的语音并进行播放。

Wi-Fi Direct

Wi-Fi Direct协议允许多个支持Wi-Fi功能的设备间进行通信，不需要控制器设备，例如热点。三星bada提供了一系列Wi-Fi Direct类，允许您管理Wi-Fi Direct设备、创建Wi-Fi Direct组、扫描Wi-Fi Direct设备、关联Wi-Fi Direct组所有者。

OpenAL

Media命名空间里的OpenAL命名空间使您的应用程序通过OpenAL（开放音频库）创建3D声音。OpenAL是一个跨平台的3D音频API，适用于游戏应用和许多其它类型的音频应用。库模型是一个三维空间移动的音频源集合，可以被空间某处的一个听众听到。

应用中的广告

从bada2.0开始，bada平台提供了一套容易使用的API，用来插入有针对性的广告到您的应用程序中。广告控件支持两种类型：图片和文本。当用户触摸广告控件时，触发以下动作之一。

- Click-to-Browser：在缺省浏览器中打开广告客户设置的一个网页。
- Click-to-Call：自动呼叫广告客户设置的一个电话号码。
- Click-to-Store：三星应用商店推出广告客户设置的一个产品页。

推送通知

从bada2.0开始，您可以在应用程序中提供各种形式的通知支持。您可以使用这些通知类型的任何组合：Ticker，Quick Panel快捷面板、Badge数目和声音。当bada设备上的应用程序等待用户数据更新时，Ticker模式通知用户。设备通知用户客户端应用程序需要处理的服务器上的数据。从用户的角度来看，似乎客户端应用程序总是活跃在后台等待通知。

更多框架特性

- 增强对Flash,、HTML5的支持
- 支持HTTP live streaming
- 集成支持JSON 分析器
- 增强剪贴板功能

开发工具

- 性能分析器

该工具提供了六个Eclipse视图：Monitor、Timeline、Call Stack、Events、Summary and Warning。Moniter视图显示了您使用性能分析器选择跟踪的实体列表。TimeLine视图提供了选定监控触发事件时间戳的图形化表示。CallStack视图显示了应用程序调用方法的顺序，最近调用的方法总是在顶部。Events视图显示了Monitor视图选择监控事件的数据。Summary视图显示了事件所使用的线程和对象。Warning视图显示了基于跟踪结果的警告。它列出了错误条件和引起错误的对象。

- Profiler

Profiler是IDE中另外一个工具，帮助您很好地调整您的应用程序。Profiler计算出应用程序方法调用的数量和执行时间，并确定花费最多时间执行的代码段。这些信息通过标识代码瓶颈进一步帮助您优化应用程序。为了得到Profiler结果，需要使用-p和-pg配置选项建立您的应用程序。当使用这些选项建立的应用完成运行时，IDE的Profiler视图显示。

- 仿真器Emulator

模拟器Simulator环境改变成仿真器Emulator。现在您可以多次调试和运行多个应用程序，不需要每次都重启设备仿真器。

Web应用程序框架

从bada2.0开始，可以使用Web技术建立整个应用程序。

JavaScript框架：可以创建丰富的用户界面，它和本地bada应用程序具有同样的外观。

HTML5：可以利用应用程序缓存，web存储，canvas，地理位置，视频，音频，WebSockets，跨文档消息和服务器发送事件的优势。

WAC API：可以使用设备API，例如PIM、Device、Messaging、Camera、Accelerometer、Orientation和文件系统。

欲了解更多信息，请参阅开发者向导和版本发布说明，详细了解每一个已经引入到命名空间的新功能。

欢迎您从开发者网站（http://developer.bada.com > Dev Tools > SDK Download）下载bada最新SDK，测试您的应用程序。

用户体验导向的 Android应用开发

文 / 陈彧堃

本文指出“流畅的环境”、“友好的体验”和“节省电量”是保证Android应用拥有良好用户体验的三要素。

Android开发目前是移动开发中的“当红炸子鸡”，大量Java程序员涌向Android，同时会习惯性地将桌面和Web端的开发/设计经验带到移动设备上。这样的好处是充分利用了移动开发和桌面/Web服务的共性，比如广泛使用的列表、本地数据库等常用组件；坏处是移动和桌面/Web的使用场景和载体完全不同，直接移植桌面端开发的经验有害无益。

比如，手机主要在碎片时间使用，用户容易对复杂的界面设计感到疲惫；同时，移动环境中上网慢，网络连接频率和失败重发机制的设计更有讲究；此外，手机电池续航能力差，后台复杂的计算会加速耗电速度。这些开发理念直接影响用户最终体验，下面我们来讨论一下在Android中如何以用户体验为导向进行开发优化。

虽然不用深入了解底层，但需要对系统有基本的了解。Android系统分层清晰，最底层是Linux Kernel 2.6，之上包含了Webkit、SQLite、OpenGL ES等基础C/C++库，同时Dalvik虚拟机运行于Kernel之上，帮助应用进行底层内存管理（这样使Android应用无法直接进行内存释放）。这些库一方面被系统大量使用，另一方面也通过Framework层提供接口给开发者。此外，Framework层还提供其他系统级的服务，如消息通知服务、位置获取服务、设备信息读取服务等。

由此可见Android对于开发者非常开放和灵活，尽管如此，开发时仍然要注意不要过于随意，以免产品过于复杂而让用户不知所措。

当然，除了少数系统级应用开发需要深入了解Framework层实现机制之外，一般第三方应用开发者并不需要深入了解每一层原理，应把重点放在如何理解和灵活运用庞大的Android SDK API。

本文主要围绕用户的三种感觉来说明如何进行开发。

流畅的环境

让用户感觉使用非常流畅。迟缓会潜移默化地留下不好的印象。用户看见App的图标，便会在心中和“迟缓”、“卡”、“不稳定”画上等号，产生“打开畏惧症”。

用户滑动Listview、Gallery、Coverflow时觉得卡，多半是因为相应Adapter对getView的处理不够好。每个Item都会和数据源绑定，而数据源的获取方式有多种：网络、本地文件、SQLite数据库、SharedPreference以及内存，它们的传输时间分别是7秒、2秒、1秒、100毫秒、5毫秒。

对于最耗时的网络请求，很多人会采用异步操作，不会让用户耗费精力在网络等待过程中。但在I/O以及SQLite查询时，用户的等待时间容易被忽略，从而降低滑动的流畅感。Android用户常常遇到的ANR（Application Not Responding），便是这个问题的升级版。要知道，Activity Manager和Window Manager监视着应用程序的响应，当发现按键或触摸发生后5秒还没执行完处理逻辑，或是BroadcastReceiver处

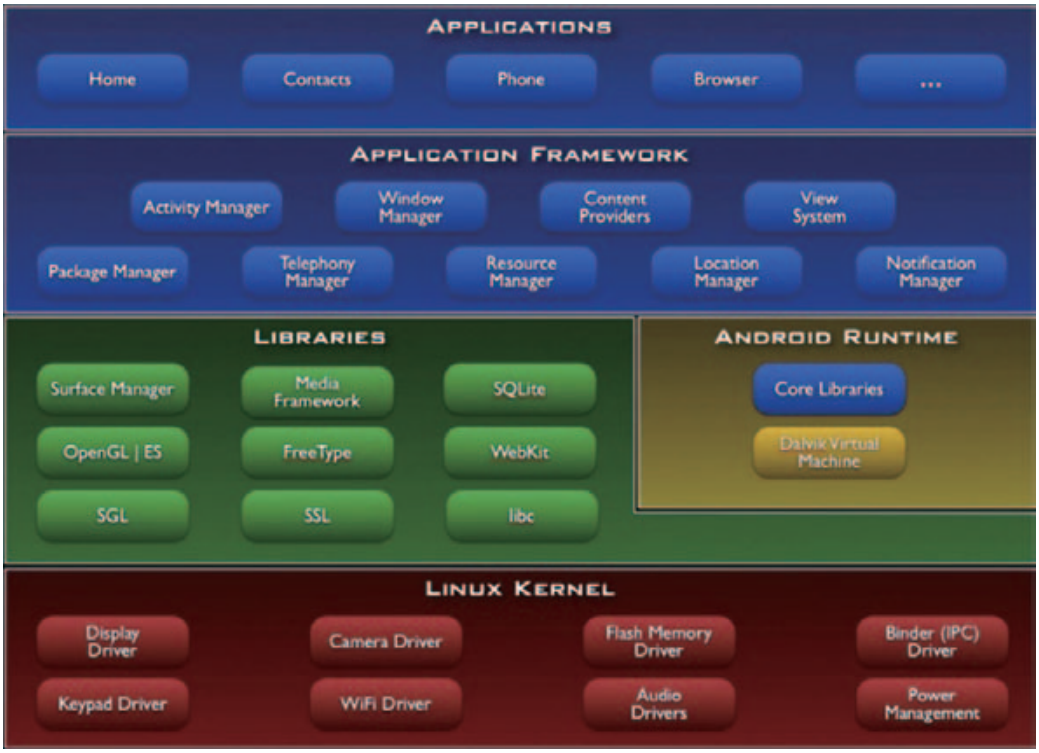


图1　Android架构图（图片来源：http://developer.Android.com/guide/basics/what-is-Android.html）

理时间超过了10秒，系统便会抛出ANR错误，并提醒用户强制终止应用。

我的建议如下：

■ 对于无法在短时间完成的操作，在独立线程中处理，Android有多种异步处理模型可供使用，包括Thread-Handler、AsyncTask以及Loader and CursorLoader。

■ 尽可能减少复杂计算和降低I/O，充分估计对象的使用频率，选择合适的数据源。个人认为大部分应用中不会存在太多太大的对象，可以考虑将数据缓存在内存中。如果应用中有太多图片不能一直缓存，可采用LRU（Least Recently Used ,最近最少使用）算法将不常用的缓存清理出内存，这样缓存大小可控，从而不会出现Out of Memory（内存溢出）的Bug。

但要注意，算法是把双刃剑，如果你享受到类似LRU带来的提速后的爽快，就可能会挖空心思探索更高效的算法。这时要慎重，后面会讲到看上去很牛的算法带来的问题。

另外，网络等待虽然是最耗时，但却容易被忽略。因为粗看上去网络是不可控的，与开发无关。一般会设置几秒钟的超时，超时则重发。事实上，在国内，中国移动的GRPS网络占主导，所以手机上网普遍很慢，HTTP连接上下行10秒是很正常的，超时设置20秒都不为过。同时，根据友盟对Android应用使用的统计，用户在每个App上的一次启动花费时间是1分钟左右，理论上有3次重发机会，但一次超时（假设是20秒）后，用户就已经失去信心，不会再等待一次了。所以在开发时，要结合具体使用场景，设计数据预取机制，尽量降低网络请求次数，同时考虑gzip、protobuf等数据压缩和编码机制，保证一次取到的数据不至于太大而造成额外延时。

友好的体验

不友好的体验来自三个方面。

其一是Android的碎片化带来了UI适配问题。Android机型众多，和iPhone相比，界面适配饱受诟病。要保证应用能运行在不同分辨率的手机上，需要理解Android提供的自动适配方案。事实上，Android系统为UI适配做了充分的考虑，只要理解系统对此的精心设计，就能在

开发时少走很多弯路，给不同分辨率的用户提供友好的呈现界面。

简单来说可做如下解释：

- 开发时避免绝对布局（AbsoluteLayout），因为这会让你的应用只在测试机上“看上去很美”，放到别处就横七竖八。
- 界面控件大小单位多用DIP（Device Independent Pixels），理由同上。
- 图片尽量使用系统提供的NinePatch技术，能使同一张小图片在不同分辨率的屏幕上保持精度自由缩放。

其二是滥用通知服务，导致用户很容易被打断。典型场景是在通知栏上的各种通知消息，有关无关的都推送，让用户感觉不适。建议是通知适可而止，除非是对用户真正有用的信息，否则最好让用户进入程序后再提示。

其三是主要来自设计师的问题，就是照搬iPhone应用的设计。Android的系统特点不同于iPhone，如程序的栈式管理机制、菜单按钮、返回按钮，从而用户的预期也不尽相同。比如我发现很多设计师喜欢抛弃返回键，模仿iPhone给每一页设计一个软返回按钮。生硬做作的移植会导致与用户预期不一致，是彻底的设计败笔。

节省电量

随时都得插在墙上充电的设备，不叫移动设备。如果你的App让用户一直守着墙角，用户也会很快把你丢到墙角。你会问：“他怎么知道我的应用耗电？”很抱歉，目前来看，Android用户中有大量发烧友和技术高手，同时系统很不客气地记录了每个应用的耗电量，于是用户偶尔会去系统后台查查耗电大户，之后会毫不客气地打开卸载工具。

所以需注意以下几点：

第一，不要绞尽脑汁设计复杂算法，不要在后台跑服务，不要网断了还不停重试。在开发一个模块前先想想会不会费电，如果会，就不要去做。代码是为了服务用户，而不是折腾用户。

高手喜欢挑战，尤其在手机上实现精巧的算法，这样能带来更强的征服感。有人曾在手机上实现了布隆过滤器（一个庞大精巧的类哈希表，多用于在服务器端如垃圾邮件查找），其内存消耗和计算复杂度都远远高于普通的HashMap，且实现并不容易。结果App发布之后，出现用户抱怨耗电量大，并且经常出现Bug，最后还是老老实实换成了HashMap。任何算法的目的都是为了服务用户，如果简单自然的方法能更好地做到这点，何乐而不为？如果真的在客户端找不到简单的算法，则需要反思——为什么在手机上需要复杂的计算？是否该将这些计算放在服务器端？

第二，不要在后台滥用Service。Android非常开放，开发者可在后台触发任何处理逻辑，肆意占用CPU和内存。一般来说，Service的目的是为了监控变化，包括系统和网络变化。系统变化可通过注册BroadcastReceiver监听控制，比如应用安装和卸载等事件，这样耗电量非常小，完全可替代在Service中轮播。网络请求无法用BroadcastReceiver监听，但是有两个建议。

- 无严苛的实时性要求，可延长轮播间隔，如6小时自动请求一次，同时时间隔可通过服务器在线更新。这样既省电，偶尔急需实时推送时也可在线调整时间间隔。
- 对实时性有要求，考虑使用成熟的推送服务，如Google的C2DM（http://code.google.com/Android/c2dm/），和亚马逊的AWS SDK（http://aws.amazon.com/sdkforAndroid/）。

第三，网络请求不要太频繁。系统组件中最耗电的是屏幕，其次就是网络。前文已经提到过，网络出错重发会降低用户体验，还会耗费电力。可通过数据预取结合数据压缩算法减少网络请求次数。

总之，在开发时我们要替用户思考是否做到了“流畅、友好、省电”，以保证App拥有不错的用户体验。



陈彧堃

曾任职于微软研究院，从事数据挖掘及机器学习的相关研究，对时空数据挖掘，普适计算等领域有深入了解。2010年4月加入友盟，开始打造友盟移动开发平台。

责任编辑：陈博（chenbo@csdn.net）

全景解读

Android UI自动化测试框架TMTS （下）

文 / 丁津津

本文从UI元素控制、测试流程、特性、框架、创新与未来六点为你全面解读Android UI自动化测试框架TMTS。

如何对UI元素进行控制

在拿到控件后，就该考虑采用何种方式提供API对其进行操作，是面向过程还是面向对象。当时我们主要从淘宝客户端的角度出发对比两者的优劣，因为要适配多种设备，不同设备上UI布局会有差别，若采用面向过程，测试用例的复用率会很低。而且Robotium是一套面向过程的API，也没有必要再去做一套类似的API。再者面向对象更贴近Android应用开发的方式，也能在一定程度上帮助测试人员理解应用的代码，于是团队决定提供一套面向对象的API。

面向对象的API需要将对控件的操作封装进控件，正如Android.view.View及其子类提供的那些API。

接下来的问题就是TMTS控件类是否需要继承Android.view.View类。Android的线程安全决定了更新UI只能在UI线程里面操作，如果在测试代码中直接操作，程序肯定会异常退出；如果直接继承Android.view.View类，会有相当数量的API需要重写，并且考虑到测试中并不会使用那么多的API，过多的API反而会给测试人员带来困惑，这也是API易用性的考虑。团队决定不直接继承View，而是采用组合的方式生成自己的控件类。

不同于应用开发，在测试中并不需要将控件类型分得很清楚，绝大部分操作都只是点击，因此没有必要对每个Android.view.View的子类都封装一个对应的TMTS控件。我对原有Android控件进行了一次抽象，基类为TmtsView，这里定义了所有控件共有的操作。对于SDK中能编辑文字的控件如TextView，EditText统一抽象为TmtsTextView，对于容器类的View统一抽象为TmtsViewGroup，针对GridView和ListView继承此类，抽象一个TmtsAbsListView，最后WebView的操作比较特殊，单独用一个TmtsWebView。

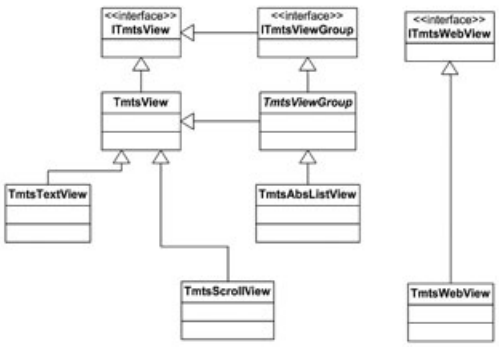


图1 TMTS控件关系示意图

在整体上设计好控件类型之后，接着就是具体实现，因为对原生View进行操作的只能在UI线程中完成，而正好Instrumentation类提供了相应的API：runOnMainSync(Runnable)，因此TMTS的控件类，通过组合View和Instrumentation得到：

```
public class TmtsView implements
ITmtsView {
…
protected Instrumentation inst;
private View view;
   …
}
```

■ ITmtsView是一个接口，定义所有控件通用的操作。

ITmtsViewGroup继承ITmts，定义了TmtsViewGroup的操作。

TmtsView实现ITmtsView接口，是所有控件的基类，所有通用操作都在此类实现。TmtsTextView继承TmtsView类，在其基础上添加了对文本的操作。

- TmtsScrollView继承TmtsView类，定义了ScrollView的一些操作。
- TmtsViewGroup类继承TmtsVeiw并实现ITmtsViewGroup接口，这是一个抽象类。其具体实现为TmtsAbsListView类。
- TmtsWebView比较特殊，独立于其他控件类，后期会将其并入ITmtsView的继承体系。

使用TMTS进行测试的流程

TmtsTestCase类是使用框架的入口类，测试人员在编写测试脚本时继承此类，其中提供了获取UI元素的API，这些API会返回封装后的TMTS类型元素，调用这些TMTS类型的元素提供的API，就能对UI元素进行控制了。

TMTS大部分API都是面向对象的，但有些操作用面向对象的方式提供出来就显得有些勉强了，比如模拟点击设备上的一些物理按键，正如老子所言：“天下皆知美之为美，斯恶矣。”这些操作都以面向过程的方式在TmtsTestCase类中提供。

TMTS的特性

- 无须import被测工程

查找View的过程通过反射实现，不再需要将被测应用的代码import到测试工程中。

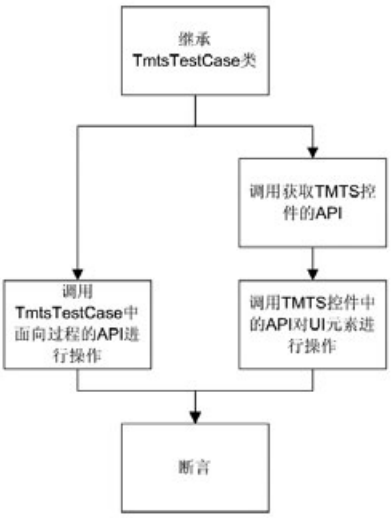


图2 TMTS测试流程

- 能自动对多个Activity进行测试
- 面向对象的API

提供一套面向对象的API，符合Java的编码风格，有助于提高测试人员的白盒测试能力和定位Android应用Bug的能力。

- 对WebView的支持

提供对WebView的支持，能够操作WebView里面的页面元素。

- 框架关键代码都加上日志，不仅可以在logcat中输出日志，也能将日志保存至设备的SD卡上，方便了日志的分析。

TMTS与其他框架的比较

对目前已有的Android UI自动化测试框架还有Robotium和Android Native Driver。

面向过程和面向对象各有优劣。面向过程的API可以在没有应用源码的情况下对应用进行测试，用起来和正常的手工操作过程一致，但复用性会差一些。

目前Android Native Driver和TMTS提供的面向对象的API在没有应用源码的情况下需要借助hierarchyviewer工具，而且应用编译没有使用混淆器的情况下才能进行测试，但面向对象的API更加符合Java编码习惯，用例的复用性也会高一些。

Robotium和TMTS都是直接以jar包的形式

表1 TMTS与其他框架的比较

特性	TMTS	Robotium	Android Native Driver
开源	●	●	●
运行方式	Instrumentation	Instrumentation	WebDriver
测试脚本使用语言	Java	Java	Any
面向对象的 API	●	--	●
不更改被测应用	●	●	--
测试预先装好的 apk	●	●	--
WebView 的自动化	●	--	--
持续集成	●	●	--
Remote Control	--	--	●
出错截屏	●	--	--
录制回放	--	--	--
性能数据收集	--	--	--

添加到工程的buildpath中，就可以在测试类中直接使用了。

Android Native Driver要稍微麻烦一些，它分两个jar包，即Server端和Client端。Server端jar需要添加到被测的应用当中，相当于要修改应用。Client端jar包放在测试工程中。目前使用还需要在命令行中手动打开设备的端口。

TMTS 的创新点

■ 提供布局文件解析工具，能按照指定的规则自动生成包含被测对象的类：在使用TMTS在获取控件时，需要使用hierarchyviewer分析当前Activity布局，获得相关控件ID，在易用性上还有欠缺。目前的想法是由布局文件生成被测对象的Java类，方便在测试代码中利用IDE代码智能感应功能，直接使用被测对象类，减少出错概率。实现方法是使用XSL解析布局文件生成Java类，将相同类型的节点变成内部类，将有ID节点变成内部类中获取控件的方法，然后再提供一个获取自动生成类实例的API。

解析布局文件生成的类全部继承自TmtsGenClass：

```
public class TmtsGenClass {
    protected TmtsTestCase context;
      public void
setContext(TmtsTestCase context) {
      this.context = context;
}
}
```

TmtsTestCase继承了ActivityInstrumentationTestCase2类，提供了TMTS中获取UI元素的API，Layout方法也定义在TmtsTestCase类中，它能返回TmtsGenClass子类的实例，并且将一个TmtsTestCase类的实例传递给它，这样TmtsGenClass的子类就能通过调用定义在TmtsTestCase类中的获取控件的API来实现其返回控件的API：

```
public <T extends TmtsGenClass> T
Layout(Class<T> t) {
      try {
            T result =
t.newInstance();
            result.
setContext(this);
            return result;

      } catch
(IllegalAccessException e) {
            throw new
RuntimeException(e);
      } catch
(InstantiationException e) {
            throw new
RuntimeException(e);
      }
}
```

■ 对WebView的支持：在这里简单介绍下WebView自动化的实现方式。如何实现WebView自动化？有Android编程经验的同学都

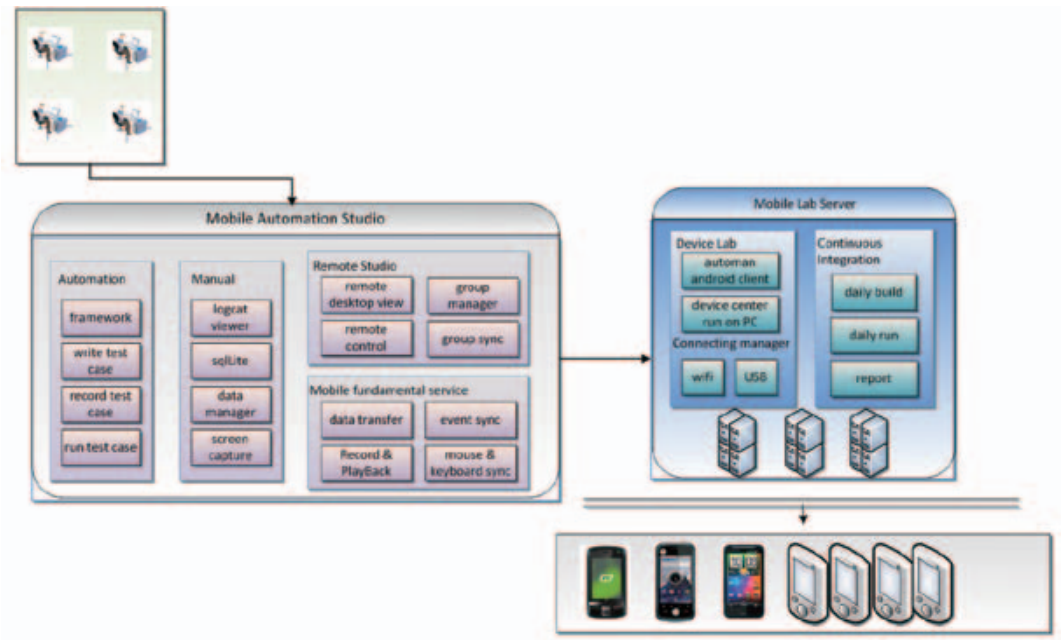
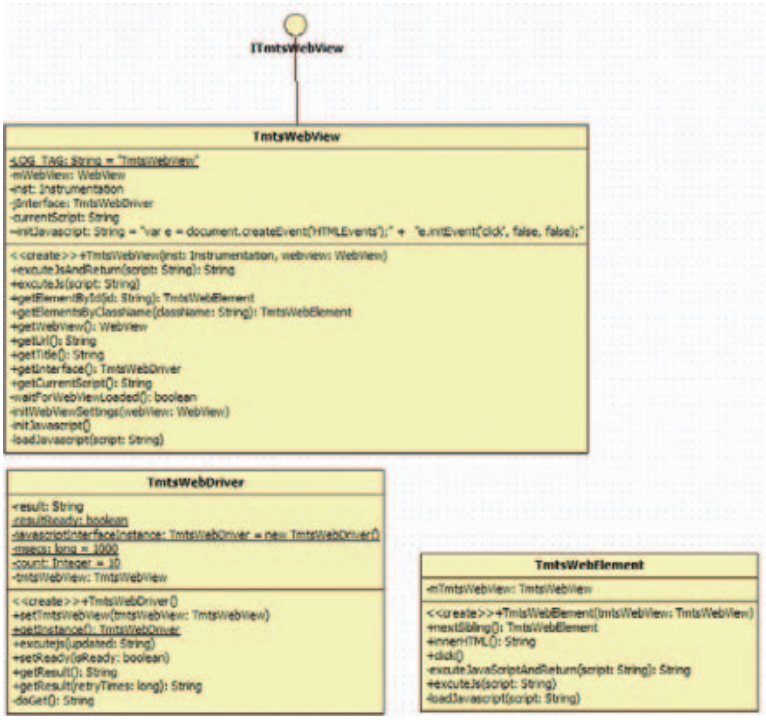


图3 TMTS长期规划



TmtsWebDriver：实现提供给JavaScript调用的Java函数TmtsWebView：实现Java对JavaScript的互调
TmtsWebElement：实现Web元素的操作

知道WebView是通过WebKit 渲染引擎实现的，也就意味着需要对HTML的DOM元素进行操作。很多经验可以借鉴淘宝UI自动化的实现方式。我们利用WebKit 的Java-JS-Bridge实现Java和JavaScript间的对象转换，从而达到对Web页面元素的控制。

在这里我们有必要对addJavascriptInterface绑定机制做大致的讲解，因为接下来的很多工作都基于这个展开。Android在初始化WebView时会由WebViewCore启动一个监听线程WebViewCoreThread。同时初始化一个消息队列，每次WebView进行一次addJavascriptInterface操作，然后将绑定对象存入消息队列。监听线程接收消息后，进行一次过滤，然后交由JNI处理。针对多线程异步绑定机制，TMTS在设计的时候制定了一个循环等待策略，主要在TmtsWebDriver类中实现。

TMTS的未来

测试人员在PC端对远程实验室的多种Android设备进行自动化测试。目前TMTS正在向这张图所规划的方向努力。其中包含近期的计划：

远程控制：类似Android Native Driver，在PC端启动测试工程，实际的测试运行在远程的设备上；

操作同步：对一台设备进行测试，其他设备获得同样的操作。

总结

测试开发不同于纯粹的应用开发或纯粹的测试，我承担过三种角色，测试框架的开发比纯粹的应用开发更能加深对Android系统的理解，会经常去查看Android的源码以了解其底层实现机制。

测试开发工程师的客户是一线测试人员，与他们经常沟通很重要。我曾不止一次与一线测试人员沟通API的使用情况，如果提供API之前不与测试人员沟通，API很可能会是失败的，通常表现为易用性差。

得益于淘宝自动化测试团队在自动化测试方面长期的积累，在具体调研之前，整个TMTS功能的结构图就已经出来了。稍微了解软件测试的同学都知道在设计阶段纠正Bug所花的代价是最小的，假如TMTS一开始没有进行足够的调研就开始实施，结果不是走了不少弯路就是失败。

大成若缺，其用不弊。世上的事情本没有一件是完美的，软件中的设计也是如此，该用面向对象还是面向过程，该继承还是组合都没有一个定论，选择最合适需求的才是最重要的，有些时候还需搭配起来使用，例如Java是面向对象的代表语言，但它也提供了static关键字。

（TMTS已经开源项目地址：http://code.taobao.org/project/view/565/）



丁津津

淘宝软件技术质量部自动化组Android测试开发工程师，从事Android自动化测试框架方面的开发和研究。

责任编辑：陈博（chenbo@csdn.net）



罗剑
赶集网技术副总裁

创业型公司：小公司，大平台

专访赶集网技术副总裁罗剑

记者 / 张蜀

赶集网已为公众所熟知，但对其技术管理者的情况则知之不多。近日，本刊记者专访了赶集网技术副总裁罗剑，请他分享自己的真知灼见。

《程序员》：请简单介绍一下你的经历。

罗剑：2001年，我从南京理工大学本科毕业，然后保送读研究生。研究生只读了一个学期，就去了新加坡，在新加坡国立大学获得硕士学位。毕业后我去了由微软和埃森哲合办的公司Avanade，专门做微软架构顾问。在Avanade，我的主要工作是为新加坡税务局一个全新的系统编写底层的函数库。2004年我回国到微软工作，服务于64位Windows生态系统，也就是帮助中国软件公司迁移软件。2005年换到MCS（咨询服务部），带领团队做政府、教育和能源相关的项目解决方案。2006年，我从微软离职，做了一个非IT的生意。一年半后，把所有股份卖给合伙人，从中抽身出来。又经过两个月左右时间的考虑，也就是2008年7月，加入了赶集网。

《程序员》：刚加入赶集网时是何景象？

罗剑：当时，我们正在和Google谈分离，谈判过程比较漫长，也经历了艰苦的阶段，感觉都快发不出工资了。但融资还算顺利，很快融到了第一笔钱，也得到了诺基亚的投资。那时的技术人员还不到十个，大家挤在一间小屋办公，到冬天时甚至都可以不用暖气。

《程序员》：既然条件如此艰苦，那当初为何选择了这样一家创业型小公司？

罗剑：当时也有别的选择，比如回微软。而之所以选择小公司，有五点考虑：第一，在大公司，每增长1级需要1~2年时间，未来5年的发展可以轻易地预见到；第二，大公司的条条框框，使得员工不能轻易改变自己的方向，转到感兴趣的部门，比如微软就需要经过两个部门的负责人同意方可调转部门；第三，在于自己的追求——不仅想在技术方面得到提升，也想在综合能力上有所发展，这便需要一个更大的平台，显然不是一个大公司的小螺丝钉能做到的；第四，在创业公司可能得到更大的物质回报，这是大公司所不能的；第五，退一步说，大公司的门永远是为优秀人士打开的，如果在小公司做不好也可以转投回去。

《程序员》：现在苦尽甘来了。做了这么长时间的技术副总裁，有什么体会？

罗剑：第一，部门经理或者部门经理的领导，都要注意关注人，要使部门同事像家人一样相处；第二，要有一个好的产品管理团队，这是互联网的特点，因为互联网发展太快，哪些是必须做的、哪些是可做可不做的、哪些是稍后再做的，需要非常明确；第三，强调快速迭代，通过参考敏捷开发，以自己最快的速度，保证每天有上线的产品；第四，在招聘员工时，要看他脾气是否与公司对口、他对公司产品是否看好、他能否与团队互相尊重，掌握好这三点，就可以招到比较合适的人。

《程序员》：能否结合赶集网谈谈移动互联网可能的机会？

罗剑：移动互联网和互联网没有本质区别，最核心的就是为用户提供价值。当然，移动互联网具有随身性、便利性、地理位置信息等特性。微博这种杀手级应用，让用户利用闲散时间，就可以发布和浏览信息。赶集网也是如此，围绕生活分类信息，让用户更快捷、更方便地发帖和看帖。此外，手机对赶集网还有一个利好：赶集网一般都是线下交易，本来就需要打电话，这样就实现了自然的串联。总地来说，一切都是为了给用户提供价值。



吴永强
去哪儿网CTO

工作就是这么简单

专访去哪儿网CTO吴永强

记者 / 高松

早在2007年，吴永强就加入了去哪儿网，全面负责公司的技术管理工作。他将在本文中分享在去哪儿网的难忘经历和技术感悟。

吴永强在大学期间主修物理专业，毕业后改做计算机相关工作，安全、平台开发等都涉及过，最长的职业经历是在雅虎中国工作的那五年，主管运维工作。作为雅虎中国的第一位运维工程师，吴永强经历了运维规模从一两千台服务器到五六千台的发展，运维方式也从靠人工夜以继日地工作到标准化、自动化部署的转变。

《程序员》：你个人在雅虎的作用或者贡献是怎样的？

吴永强：在这样的大公司里，谈贡献有点奢侈，对我来说学习的成分更多些。雅虎的经历给我一些很好的启发：第一，技术一定要和自身业务是匹配的，不管什么先进的技术，关键是对业务要有价值；第二，技术的选择一定要跟公司的业务和员工相匹配。这段经历让我变得非常务实。

2007年，吴永强加入初创的去哪儿网，全面负责该网站的技术工作。刚进入去哪儿网，他遇到了诸多问题。“首先是成本，我记得很清楚，当时买一台服务器我要考虑很久，机房和带宽的成本也需要考虑；其次应用开发、部署速度慢，不能满足业务的需求，”吴永强谈到在资金压力下，开展业务面临很多困难，“而且互联网初创企业的业务增长根本没有计划性，功能扩展很快。”从老板到员工，大家都很着急，都想把业务快点儿做起来。如何解决这些问题，对吴永强提出了很大的挑战。

《程序员》：来到去哪儿网时，你对它整个技术的看法是什么？你是怎样做的？

吴永强：那时去哪儿网的规模非常小，产品还是一个雏形。我来了之后先看看这个团队每天在干什么，一个礼拜之后，我就明白它最大的问题在哪里了：一个是稳定性，另一个是性能。

稳定性是个架构的问题，这跟去哪儿网的业务特点相关。去哪儿网的业务天然地要求使用非常多的AJAX不停请求后端服务器，所以并发量很高。其他网站十万访问量用两台机器就够用了，而我们需要更多还搞不定。所以用传统的方法去解决这个问题，显然是不合适的。当时的系统老撑不住，性能也不是很好。工程师们每天不停地重启系统，不停地加日志研究哪里慢了，有限的人力和精力都被消耗在处理这些问题上面。

所以我当时做了几样事情，第一件就是更换有问题的系统模块。以前使用的NFS共享存储结构有目录缓存延时，会造成很多问题，性能很差，所以我们换成了memcached+NFS的结构。系统稳定性就有了改善。同时每一个请求服务线程生命太长，也会导致系统压力过大，于是我们将结构拆解了一下，将最容易恶化的用于获得报价的不受控的线程换成一个受控的线程池，并最多控制在十个。虽然有0.01%的请求因线程池限制获得不了数据，但由此换取的是系统变得非常稳定。

解决了上下关心的稳定性和性能问题，吴永强开始着手开发新的产品功能，支持新的业务。然而新问题出现了，当时去哪儿网的工程师都是做Java的，对于Web应用的特殊性不敏感。他举了内存管理的例子：“那时Java工程师非常喜欢在内存里保持Cache，而且都是无界的，运行时间一长，经常造成OOM。这时我就要求每个工程师

都要会计算系统在当前压力以及未来流量增长一倍时需要多少内存，并要求对线程数量和CPU使用率也能使用数字来计算，这样系统基本就掌控住了。”如此一来，开发进入了一个比较正确的轨道。2007年底，去哪儿网的用户量开始飙升。

《程序员》：用户量增长带来什么问题？

吴永强：随着业务的增长，每天都会有新的问题，但我们有了一些自己的基本架构，以及培养了一批富有经验的工程师，他们能够在这些结构上不停地作调整，进行改进来满足系统规模扩展的需要。所以我有更多的时间作为技术管理者的角色来开展工作。在管理中，我尽量给工程师们一个能够发挥他们能力的工作环境。但有时，很多新的做法也不能被工程师马上理解，也会迫使我强力推进一些做法，取得效果后再建立共识。

《程序员》：你如何总结CTO角色？

吴永强：我觉得很大一部分工作是如何协调产品和技术的关系。产品和技术永远是一对矛盾，因为产品创造需求，技术实施需求。这两个部门有自己天生的倾向，业务部门更倾向于多变，特别是互联网公司；技术则要求稳定，抵抗变化。我总结：两者应该保持既竞争又合作的关系。合作是为了服务业务，竞争是看谁对业务的贡献大，谁的影响力更强。

在“去哪儿网”我经常强调节奏比什么都重要，做互联网产品开发一定要有自己的周期，就像计算机一样，总得有晶振来保持频率，周边的系统才能围绕这个频率进行工作。通常互联网公司都是以产品开发为核心，这就要求技术部门有一个固定的频率接受需求，发布产品。其他部门也就能明确知道自己的需求什么时候能被采纳、被执行。这样公司的运作才能够变得比较有序，这种方法在“去哪儿网”实施后，效果是非常好的。

《程序员》：对负责系统方面技术管理的同行有何种建议？

吴永强：对现有系统我基本上不赞成完全推倒重来。当然我并不反对重构代码 。对于创业公司，往往公司的很多系统是由一个工程师写的。随着业务、产品和团队规模扩大以后，必然就会有一个逐渐移交的过程。这时接手的工程师最常见的是， 总觉得人家代码写得烂，需要推倒重写。

包括我在内的许多工程师都会这么做，这很正常。因为系统架构会因为业务发展和复杂化逐渐变得不合适，所以大家喜欢停下来重构系统。但在去哪儿网我们基本不这么做，我们的做法是，继续按周期发布产品，每一个版本的发布里都有对原来结构的一点改进，重写一小部分代码，这样坚持一年之后，你会发现系统相当于重写过一遍了。这样做比较容易控制风险，而且不会影响业务发展的速度。

这么做的原因有二。其一，推倒重来是非常影响业务的，不到万不得已，最好不要这样做。譬如Myspace后来一蹶不振，其中一个重要原因也是因为他们停下业务来化很长时间去重写他们的系统。 其二，要选择风险最小化的做法。一个有年头的系统，通常有非常多的业务逻辑在里面，如果要做一次完全重构，投入的资源会非常多，而且风险也会很大，对于初创公司并不合适。

《程序员》：对去哪儿网技术的发展有什么样的展望？

吴永强：去哪儿网服务的是一个垂直领域，所以我们的系统比较多，牵涉的技术也很广，既有全网网页搜索系统，又有垂直实时搜索引擎；我们还有UGC系统，譬如问答、博客等；另外还包括交易系统，如给机票供应商使用的或是给酒店供应商使用的系统。这些系统之间的技术跨度是比较大的。有限的工程师要支持这么多不同结构和特点的系统，对我们的挑战非常大 。其次，目前我们80%以上的工作都是信息的收集、产生、处理、展现。现在我们能够收集和产生非常多有价值的数据，包括酒店预订、飞机、火车票以及各种景点、攻略等数据，如何将这些数据进行处理、整合，为用户提供更多的价值，目前业界并没有太多人在做，但我觉得这将是去哪儿未来发展的重要方向。

被问到事情做成了是否感到一种成就感时，吴永强说：“成就感就是一瞬间的，绝大多数是压力，我觉得创业公司肯定是这样的，可能一年有360天有压力，只有5天是很有成就感的，工作就是这么简单。”

构建开放的软件团队文化

文 / 许正华

伴随着软件开发敏捷化趋势的发展，“以人为本”的软件团队建设理念越来越得到广泛的重视，但在这方面的研究、探讨与实践则显得乏善可陈，现在是到了个体、团队、组织甚至是社会力量一起行动起来的时候了。

构建开放的软件团队文化绝不仅仅是搞几次技术交流和设计讨论那么简单，它是一项长期的系统工程。开放的软件团队意味着什么？个人与团队应该如何行动？作为团队的管理者以及团队成员都应该认真思考这些问题。

一个程序员的文化观

文化的自然属性

“文化”总给人一种高高在上的感觉，这多少是受到狭义文化定义的影响，即文化是人类所创造的语言、文学、艺术及一切意识形态在内的精神产品。在中国古籍中，“文”既指文字、文章、文采，又指礼乐制度、法律条文等；“化”是“教化”、“教行”的意思。

文化是可以培育、教化出来的吗？我们可以从文化的产生与传承略加分析。一种文化的产生必定源自人类内在与外在的需求，其传承少不了培育与教化的功劳，但也要遵循事物发展的客观规律。因此，我认为一个有生命力的文化必须是顺应自然的。今天，我们的很多传统文化面临着生存的危机或者被边缘化，主要原因就是那些原本支撑它们的内外需求已不复存在，这些文化必然会被历史所淘汰，只是个时间问题。

文化的封闭属性

构建开放的团队文化是一项艰巨的历史工程。很多人可能没有意识到“开放”与“文化”是一对矛盾体，因为文化从本质上有着封闭的属性。弱势的文化想尽各种方法避免被强大的文化所同化或吞并；强势的文化则希望尽可能地扩展以获得更大的利益。从这个角度讲，人与人之间的战争、组织之间的吞并都是文化现象，是人类文化的一部分。

人—文化的综合体

我们讨论文化一般都以群体组织为背景，其实每个自然人都是一个复杂的文化综合体，包含语言、艺术、科学、信仰甚至宗教等多种因素。这个文化综合体也没有超越前面讲的两个属性。

首先，我们绝大多数人都在相似的教育制度里接受着同质化的教化，但最终在文化上的表现却千差万别，这说明个体文化是一种自助餐式的文化：餐厅能够提供什么样的食物可供选择决定了我们可以吃到什么，但最终吃到肚子里去的却是我们自己选择的结果。

其次，每一个独立的个体都难以逃脱封闭的趋向性，只是程度不同罢了。这种趋向随着年龄的增长或者知识与经验的增多，得到不断的增强。一个典型的例子就是，人年纪越大其生活习惯越难以改变。我就曾经出于健康的原因试图劝说父母在某些生活细节上做出适当的调整，但最终都以失败而告终。最后，当我意识到这可能与文化封闭性有关时，也就不再做这种无用功了。

什么是开放的文化

开放的文化是一种文化表象，其实质与民主、自由、平等、容忍等人生理念与社会理念有关。谈到民主与自由就不得不提及胡适，这位中国新文化运动的领军人物。下面两段引文虽然是在讨论民主，实际为我们勾勒出了一幅以“开放”命名的画卷。

引文1：“胡适所说的个人主义的精义，可以概括为‘独立自主，参与奋斗’八个字。民主的生活方式，就个人来说，是独立的、是有充分个人自由的；但就整体来说，是参与的，是有一个共同理想的。独立而不参与固然无法达成这个理想，参与而不独

立，也会失掉个人的价值。在民主的生活方式中，中国人所最需要的是参与而不党同，独立而不避世的精神。”（《少不读鲁迅，老不读胡适》）

引文2：“民主的生活方式，在政治制度上的表现，好像是少数服从多数，其实它的最精彩的一点是多数不抹杀少数，不敢不尊重少数，更不敢压迫少数，毁灭少数。”（《胡适手稿》）

胡适的声音今天听起来仍散发着无尽的光芒。在一个开放的组织中人人应该是平等的，至少在精神层面。我很难给开放的文化下一个简单的定义，但至少可以从两个方面来甄别“真开放”与“假开放”——自主性与双向性。在一个真正开放的组织中，信息的交流是多样、自由的。任何屈服于权力的开放都不能持久，也不能深入。在一个真正开放的组织中，信息的交流是双向的。一位站在讲台前滔滔不绝的演讲者与坐在台下虔诚的听众都不是开放文化的代表。

开放文化的一个核心与三个方面

两种或者多种文化间的开放是建立在共同利益基础上的。如果我们把团队看作是一种文化共同体，那么建立开放的团队文化的核心就是确立共同的利益。

对于任何一个组织，“人/过程/产品”都是三个紧密相关的组成要素。接下来，我们将探讨开放的文化与三者间的关系。

人的因素

一个文化在其形成及发展的初期是比较开放的，因为它要从外界吸收充足的营养。随着它的逐渐成熟，其封闭性就表现得越来越充分。对个人而言，也遵循这样的普遍规律。不同的是，人的成熟更多地受主观意念所左右。现实中很多博学的人，仍可以保持旺盛的求知欲，来支撑其个体文化的不断发展。

我们习惯于将一个人的性格打上“内向”或“外向”的标签，那么开放的团队文化是否要我们更加外向吗？如果真是这样，开放的团队文化将变得可望而不可及。一个渴望学习的人最容易做到开放——这里所说的开放并不只是针对人，也是针对一切有用的知识。因此，构建开放的团队文化最终转化为了建立学习型的组织。

虽然构建学习型组织仍然是一个具有挑战性的任务，但至少已不是一个陌生的话题，本文对此只做一点分析：持久学习的动力来自于一种高尚的理想。**我们不能只满足于完成自己所能的事情，而要致力于完成自己所希望的事情**。

过程因素

也许很多程序员会质问我为何要将过程硬扯进来，它与开放的团队文化何干？我可以理解程序员多少对各种软件开发过程抱有一些敌意，因为他们把很多问题都归结于过程的错。这多少与团队缺乏对过程统一的理解有关。过程本应成为程序员的伙伴，而不是绊脚石。

过程的建立与实施是有成本的，但同时也是有收益的。我们不仅要仔细规划团队在每一个过程上面的投入，也要评估相应的产出。这种评估并不容易，因为缺乏一种简单而一致的标准。

统一的过程是团队沟通的平台，是开放的催化剂，是建立开放团队文化的重要基石。

产品因素

建立一个完善的产品知识库是构建开放的软件团队文化的重要保障。试想，如果对产品知识的掌握上水平相差悬殊，团队成员便很难开展有效的沟通与合作。在我看来，没有哪个高技术工作在产品知识库上遇到像软件开发一样大的挑战。缺乏足够有效的技术文档是大多数软件开发团队所面临的实际情况。

结束语

建立利益共同体是开放团队文化的核心。激励团队建立学习的习惯、持续地改进过程、建立完善产品的知识库是具体的方法与工具。

尽管本文内容并不可以直接用来解决具体的问题，但希望能够对探究此问题的本质多少有所帮助。在采取行动前，理清概念、定义问题、分析本源总是大有裨益的。



许正华

目前就职于CA Technologies中国技术中心，从事数据备份和高可用企业软件方面的开发和研究，并专注于软件工程、知识工程、团队文化等领域的研究与实践。

责任编辑：董世晓（dongsx@csdn.net）

在中国应如何改良Scrum框架

文 / 吴穹

Scrum在中国得到了越来越多的应用，但仍存在一些缺陷，本文在总结Scrum框架缺陷的基础上，讨论应如何改良Scrum框架，以保证实施成功。

我曾在CSDN博客（http://blog.csdn.net/adwu73）上发表了一系列“为什么纯粹的Scrum在中国很难落地”的文章，其中通过解读新版的Scrum Guide来分析如果在中国严格照搬Scrum会遇到哪些困难。而本文中，我将在总结Scrum框架缺陷的基础上，讨论应如何改良Scrum框架，以保证实施成功。

何为改良Scrum框架？Scrum Guide开宗明义：Scrum不是一个具体技术或流程，而是一个可以与其他流程和技术相结合来保证项目成功的框架。因此，将Scrum与其他技术相结合不是改良，而是Scrum预设的用法，就像在装修好的房子里再摆上家具一样。而所谓改良就是修改Scrum核心框架的内容，就像打掉屋子的一面墙一样。Scrum框架的核心内容包括团队的**角色定义**、**时间箱**、**交付物和规则**，下面我们就一一来看应如何改造。

在改造之前，同时也需要了解哪些框架是Scrum的本质，因为如果修改了这些本质内容，就好像装修不能打掉承重墙一样，不是改良Scrum，而是推翻Scrum。Scrum的三个支柱是**透明**、**审视和适应**。透明强调团队之间通过透明来加强新人，应对变化；审视和适应是指在过程中，通过不断审视来不断进行适应性改善。好了，找到了承重墙，避开它们，然后就开始干吧！

角色定义——Manager/Lead

Scrum当中设定了三种角色：Product Owner、Scrum Master和Team。Team当中由参与交付的Developers、Testers等角色组成，注意这里是没有给Manager和Lead留出位置，因为Scrum框架当中隐含了团队应该是自管理的要求。这其实和欧美的实际情况有关，在那里你能看到大把的白头发程序员，而在国内，则非常稀少。

在我帮国内客户实施敏捷的过程中，发现许多组织由于研发团队都很年轻，并不敢真正实施自组织，而Manager或Lead又没有了位置，于是纷纷伪装成Scrum Master，然后一切照旧，而真正该做Scrum Master的人反而没有位置了。这造成了许多Scrum实施的失败。其实，管理人是最难的一件事情，自组织只是一种可能的管理方式，而这种管理方式恰恰是在中国现在的软件企业里很难成功的管理方式。因此，我认为应该改良Scrum框架剥离自组织管理方式，而去强调Manager as Coach。

“Manager as Coach”这一思想可以在敏捷的另一个流派——精益（Lean）当中找到而且是Lean的基础，我觉得这种思维很可能更适合中国的现状，因为大多数团队中的成员都非常年轻、缺乏经验，因此，他们需要的很可能并不是自组织，而是真正能Coach他

们的Manager或者Lead。即使现在Manager和Lead还不称职，那么企业的重点也更可能应该是尽快提升这些Manager和Lead的Coach能力，转变他们的思想，而并不是去推行自组织。

还有一点需要注意，这里说的Manager/Lead，不是指那些只管人不干活的。Lean的思想建议，Manager/Lead必须是有动手能力的领域专家，这样才能真正能做到Coach一线研发人员，否则就只能纸上谈兵了。另外，最近在微博上的讨论中，大家往往将信任和自组织画上等号，其实。有Manager、Lead并不代表不信任，这种非黑即白的看法是错误的。

综上所述，我建议：团队在应用Scrum框架时，如果认为应该采用Manager as Coach这种管理方式，那么就应该在Scrum框架的角色定义当中，明确加入Manager/Lead这个角色，也可以修改团队（Team）的定义，将Manager/Lead也作为团队的一部分。

活动——Time Box

Scrum框架的另一个大问题是对架构的忽视。在中国，现在最大的问题不是过度设计，而是设计不充分，而Scrum更助长了这种轻视设计的风气。当然，强调架构设计并不一定意味着要写很厚的架构设计文档，或者进行复杂的UML建模。如何进行架构设计，做到什么程度，应有团队自己决定。下面来看一下应该如何改进Scrum框架。

Scrum框架当中的活动有几个：Release Planning Meeting、Sprint Planning Meeting、Sprint Review、Sprint Retrospective和Daily Scrum。

Release Planning meeting基本上是一个目标计划会，与架构设计无关。而在每个Sprint的Planning Meeting上，会有两个部分：第一部分澄清需求，第二部分进行设计。但时间太短，往往无法承载架构设计。

因此，严格照搬Scrum框架，在一些大量应用现成框架的产品开发过程中或一个产品的维护阶段，还可能成功。但对于大型复杂产品开发而言，不进行架构设计，结果基本上是灾难性的。

综上所述，建议团队在实施Scrum的过程中、在Release Planning Meeting之后，增加一个Release Architecture Design Workshop来进行架构设计，时间长度和Release Planning Meeting一样。当然，这个Workshop和Release Planning Meeting一样，也是可选的。

交付物——Artifacts

Scrum框架中的交付物有四个：Product Backlog、Release Burndown、Sprint Backlog和Sprint Burndown。Product Baclog比较简单，就是所有需要做的事情。这里特意没有明确Product Backlog Item是以什么方式定义的，以便可以集成这方面不同的实践（如用户故事、用例、FDD等）。

而在Spring Backlog的内容方面则有些问题，不太一致，细节就不在这里展开了，具体大家可以去看我的博客。我赞同最后在Release Notes当中的提法，Scrum Backlog就是由本迭代希望完成的Product Backlog Item组成，这些Item可以进一步拆分成任务来追踪。

改良总结

综上所述，可以将改良之后的Scrum框架总结在一张图里面，如图1所示。

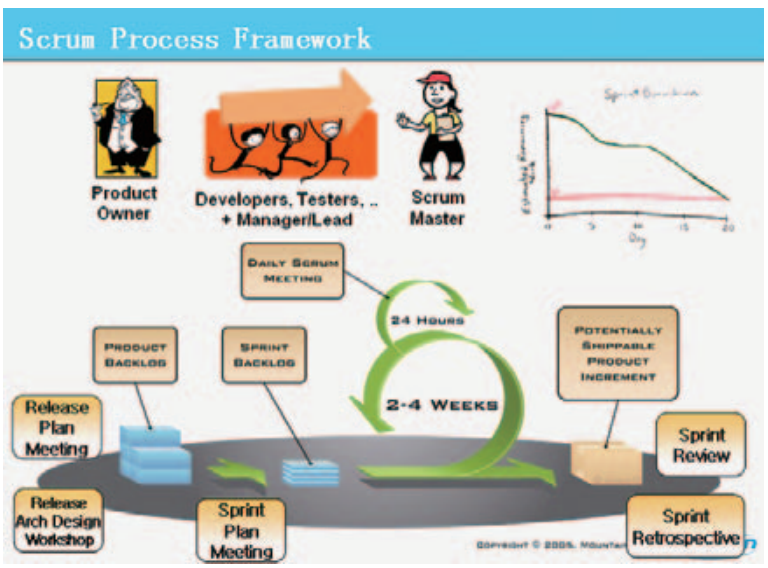


图1 改良之后的Scrum框架

实施Scrum应保持的态度

到这里，我想大家对Scrum框架应该有一个初步的了解。在正式实施Scrum框架之前，首先需要看透Scrum框架的商业本质，对它抱一个正确的态度。

Scrum框架其实可以被理解成是敏捷的商标，就像一个果农要卖苹果，卖一两个卖就卖了，但如果要卖得多，那就需要有一个商标了。Scrum框架也是一样，许多敏捷大师都有自己的敏捷理念，很难统一。因此，大家选定了这个形而上、空空的Scrum框架作为共同的商标，这样哪个大师也不会反对。

于是，你去上一节Scrum Master的培训，得到了什么？一个证书、对Scrum框架的基本了解和许多培训师自己的敏捷实践经验。因此，能不能值回票价完全取决于培训师的个人能力，而那张证书却不一定能证明你就是一个合格的Scrum Master。

因此，在实施Scrum时，不要抱着过高的期望，不要期冀银弹，而要抱着务实、开放、轻松的态度，根据企业的实际情况定制或扩展Scrum框架。

如何在中国成功实施改良Scrum框架

根据中国的实际情况，改良Scrum框架已经强化了管理者的作用，强化了架构的作用，但在中国实施改良Scrum框架还是会经常遇到一些阻碍，针对这些障碍，则必须克服才能取得成功。

■ 职能性组织、模块组织：由于实施CMMI的影响、团队年轻、人员流动等各种原因，许多企业都慢慢转向用职能性组织（如开发部、测试部、架构设计部、需求分析部或者前端开发组、应用服务器组等），实施改良Scrum框架，可以暂不改变现有的职能性管理架构，但必须要求Scrum团队的成员是跨职能、跨模块的，而且是在一起办公的，这个要求不可妥协。

■ 客户参与信任关系的建立：在国内，有时客户不太愿意参与到Scrum过程中，参与了有时还不太适应他们手中排优先级的权利，这可以慢慢培养，但必须要求客户和开发团队一起办公，共同参与Scrum过程，这样才能逐步建立起互信的关系。这在软件开发当中非常重要，因为所有估算方式都只是一种拍脑袋的方式而已，只有让和客户建立起较强的互信关系，才能共同面对不断变化的开发交付过程，而没有比一起办公更透明的方式了。

■ 变革之心与QA的正确作用：在国内，交付压力都非常大，要求团队既能交付又要不断思考可以如何优化和变革是一个非常高的要求。我认为，这里应该正确发挥QA人员的作用。QA需要放弃那种只是简单比对流程要求、不符合就开NC（Non-Compliance）的粗暴做法，而真正做到参与到Scrum流程当中，冷静旁观，不断挑战现状、发现问题、和团队一起探索解决方案。因此，在我心目中，好的QA其实是Scrum Master的理想人选，有了变革的驱动者，每个迭代的回归会议才可能真正落到实处。

■ 增量交付价值：在国内，看到许多号称做Scrum的团队每个Sprint都在交付功能点，但这些功能点并不能真正运行，这其实是伪Scrum，很可能还不如不迭代。因此，做好Scrum，就必须坚持不断交付客户可见的业务价值。也就是说，交付的代码必须是可执行、可以给客户演示的。对于复杂系统而言，这对划分Backlog Item提出了很高的要求，但是可以解决的。

■ 测试的全程参与：以前看到另一种伪Scrum，号称开发在前、测试之后一个Sprint。这种做法其实完全破坏了Scrum团队，开发测试无法形成合力。因此，必须坚持A-TDD和验收测试自动化，这又是一个很大的话题。

因此，大家可以看到，Scrum虽然简单，但真正做好Scrum需要在组织、人力、技术上有大量储备才行，即所谓“功夫在棋外”。最后，愿大家掌握改良Scrum框架的精髓，灵活应用，在实际工作取得好的效果！



吴穹

雅各布森中国公司总经理，师从杨芙清院士，有丰富的软件工程实施经验。

责任编辑：董世晓（dongsx@csdn.net）

【译者注】硕士毕业时，同学投递简历的对象莫不是大企名企，即便没能成行的也纷纷进驻外包公司，以期为最终进入外企搭建跳板。而身边一位以敏捷技术见长的同学，在一家IT巨头待了一年之后即跳入一家初创小公司，我对此颇有不解。而读过这篇文章之后，再想想这位同学非但没有荒废反而日益精进的开发技术，不禁开始钦佩他的远虑及睿智。

毕业选择起步型企业的五大原因

文 / Eric Stromberg　　译 / 李铮，李世明

近来，我收到数量惊人的E-Mail问我为什么从杜克大学毕业后会决定进入一家刚起步的小公司。而其中许多人现在都面临着相似的抉择：要么是临近毕业在选择成熟企业还是初创公司之间游移，要么是毕业后去了一家大企业工作正思索是否应该做一些转变。有趣的是，绝大多数都倾向于加入起步型小公司的职业道路：看起来他们只是希望有人能够彻底说服他们这是合理的举动。朋友和家人往往都疑惑不解：“怎么可能辞去在摩根士丹利的工作去一家只有十个人的小公司？”希望这篇帖子能够给那些希望加入起步型小公司的人一些有用的观点来回答怀疑者——为什么说在职业生涯中尽早加入初创公司是一个好主意。

首先需要澄清的重要一点：尽管我说每个人都应该一毕业就进入一家科技型初创公司，但我并不认为做起来会像说得那么容易。不同的人有不同的激情，我并不希望把我个人的观点强加给别人并笃定我所做的就是正确的。所以我想给的第一条建议就是去追求你为之激情的目标。

什么是激情？我将借助Steve Blank最近在费城大学的毕业致辞来定义它。吸引我眼球的是这句话：

“是求知欲和热忱令你为人瞩目并让你的生活多姿多彩。(It's your curiosity and enthusiasm that will get you noticed and make your life interesting.)”

我想他已然切中了要害：激情就是热忱再加上求知欲。这促使我给出职业生涯应尽早加入一家初创公司的五大原因。

激情！终极竞争优势

当我还在华尔街实习时，环顾四周的办公室我不由得陷入沉思：“我的竞争优势是什么？每个人都是那么的精明干练而且兢兢业业，不是吗？”我很快发现那些胜任本职工作的人都对金融市场充满了激情。虽然我觉得金融是有些意思，但跟同事们相比我对它并没有足够的热忱和求知欲，所以在华尔街的世界里我并没有竞争优势。但对技术和初创公司我是有激情的。如果可以在一家正在成长的公司工作，我将会用它作为我的优势。

一个验证你对现有工作是否有激情的绝妙方法就是问自己：你的周末看起来是否跟平常十分相似？下班回家后你是否发现自己会想继续工作？当听到别人讨论某个相关问题时你的耳朵是否马上就竖了起来？你是否感觉强迫症似地发问并认真倾听答案？这就是激情，它承载了如假包换的竞争优势，并且驱使你工作，哪怕在并不需要这么做的时候；驱使你思索新招儿来对付顽固问题，哪怕是当别人拉下了“工作”闸刀都在欢度周末的时候。

成长！初创公司俩月，成熟企业一年

在刚起步的公司干上一年相当于在其他任何地方工作七年。《黑客与画家》一书中Paul Graham写道：“你可以通过加入初创公司来将你的整个职业生涯经济地压缩至些许几年。与其浑浑噩噩地撞四十年钟，不如全力以赴地打拼四年。” 对于一个摩拳擦掌的年轻人来说，

初创公司可以为其职业发展提速供给难以匹敌的机遇。很自然的，在高速成长的初创公司里，工作总是比员工多。这就必然造成大量的员工“伸展”。我的意思是说你会被安放在一个也许你还没准备好的位置上，而这却是最好的学习途径。

公司总是会在最初阶段有最多的变化（与之相反的是，步入正轨后的公司则会按照既定的模式运作）。作为一名早期员工，你将会在这种演进的前排就座。因为你的角色，甚至或许是这家公司在市场上的角色，都会在快速的步伐中做出调整。于是在一家初创公司工作就会为你带来无可估量的额外收获。

轻装！历史包袱只随时间增长

我碰到的很多人都会辩解说，进入一家大企业能够降低职业风险：如果能在简历上镀层金，之后再加入初创公司即便失败，也还可以凭借之前的镀金东山再起。这里有一个被反复提及的问题：“我应该什么时候把搜集到的筹码变现去做一些我真正想做的事情？”事实上，在同其他职场里年长朋友的交流中我所发现的是，你在一条路上走得越远就越难转向。职业惯性最终会压倒一切，特别是在非科技领域内累积的技能几乎无法转移到科技型初创公司中去。

创业！最佳学习场所

我很清楚自己希望在将来的某个时刻成为创业者。这种向往能够借助在初创公司工作将抽象变为现实，但工作在初创公司所带来的影响却着实难以描述。在Hunch公司工作之前，我一直琢磨着创立一家公司，但无非是一种“纸上谈兵”，我所考虑的所有事情都仅限于理论上。而在一家刚起步的公司里工作，可以让我观察这家公司的日常运转，同时帮助我从具体的视角来审视如何创办自己的公司。应该雇用什么样的员工？应该设置怎样的企业文化？如何令新员工高效上手工作？对这些问题我还没有完美的答案，但这些事情是困扰我多年而在加入Hunch之前我甚至都不知道该问些什么的。

所有这些都源自一个事实，那就是在一家初创公司，你能更多地接触并近距离观察创业团队的一举一动。如果是在一家大企业，那些我本可以在Hunch每天一起工作的人搞不好会跟我相隔好几个管理层。我敢打赌我所学到的东西多半来自近距离的观察，如果你只是坐在自己的小隔间里而不是每天跟那些将公司塑造成型的人打交道，你也很难学习如何成为一名创业者。

阅历！多样而丰富的历练

在一家大企业，你的角色往往是已经设定好的。你所体验的分析师计划（Analyst Program）与去年的伙计所参与的完全一样。但在初创公司却不是这样，很多时候都是你来定义自己的角色，而且没有两个人会有同样的经历。如果你看到一个问题并找到了相应的解决方法，那么你很可能也就全权负责处理这个问题了。而这正是关键所在：你将全权负责。没人会精确地告诉你，你的工作该怎么做，亦或是拉你一把。而在大企业的相同位置上，职场菜鸟们就不大可能会获得同样的职责历练了。这种观念对一些人来说是令人无比兴奋的，而对那些终生在巨擘名企中腾挪的人来说则是超级恐怖的。如果这一想法鼓舞到你，那么它将带给你难以置信的回报，无论是在手头工作方面还是在学习方面。

结语

很显然，在这里我是带有无数偏见的，因为这就是我所选择的路。同时我也确信加入成熟大企业一样会有很多好处。但如果你恰好想要加入一家刚起步的公司而只需要额外助推一把，希望上述观点能够给你一些火力支援来回应你的朋友和家人，告诉他们为了一份小公司的工作而放弃一个金字招牌的职位并不愚蠢。

（原文链接：http://estromberg.com/post/6108833062/5-reasons-to-join-a-startup-after-graduating）

Hypertable：BigTable的高性能实现

文 / 郑付联

Hypertable和HBase均为BigTable的开源实现，本文从随机读吞吐、随机读延迟、随机写、顺序读、Scan五个方面对它们进行了对比测试。

Hypertable是一个开源、高性能、可伸缩的数据库，它采用与Google的BigTable相似的模型。在过去数年中，Google为在PC集群上运行的可伸缩计算基础设施设计建造了三个关键部分。

第一个关键基础设施是Google File System（GFS），这是一个高可用的文件系统，提供了全局的命名空间。它通过跨机器的文件数据复制来达到高可用性，并因此免受传统文件存储系统无法避免的许多失败的影响，比如电源、内存和网络端口等。HDFS是GFS的一个开源实现，实现了GFS论文中提到的大部分功能。

第二个基础设施是名为MapReduce的计算框架，它与GFS紧密协作，帮助处理收集到的海量数据。Hadoop是开源的MapReduce计算框架。

第三个基础设施是BigTable，它是传统数据库的替代者。BigTable让用户可以通过一些主键来组织海量数据，并实现高效的查询。Hypertable是BigTable的一个开源实现，并进行了一些改进。类似的，HBase是BigTable的另一个开源实现，与Hypertable相比，它主要是通过Java语言开发的，一些设计理念也有所不同。

系统设计

Boost C++

■ 内存密集型

Hypertable缓存所有的更新在内存数据结构（例如STL库的Map结构）。这些内存数据结构根据内存阈值或周期性地Spill到磁盘。当这些Spill磁盘文件的数量超过阈值时，它们Merge到一起产生更大的文件。Hypertable系统的性能很大程度上取决于可用内存量。可用内存的减少会导致更多的Spill和Merge操作发生，从而加重了网络和底层DFS系统的压力。同样的，额外的Heap-Merge操作也会增加系统CPU负载。对于内存需求大的应用来说，Java并不是一个好的选择，尤其是管理较大的内存Map或Key/Value对时，Java的内存性能比C++差2~3倍。

■ CPU密集型

Hypertable是CPU密集型的应用，这体现在几个方面。

首先是内存中记录Key/Value对的Map结构。转化和管理这些Map消耗大量CPU资源；此外假设Java管理这些Map使用了额外的内存，处理器的Cache也会因此变得更加低效。

Hypertable是CPU密集型应用的另一个影响因素是压缩。所有插入Hypertable的数据将至少被压缩两次，平均压缩三次：一次是写数据到Commit Log，一次是在Minor Compaction期间，还有一次是每次Merge和Major Compaction时。解压缩的数据量则是依赖查询负载的量。Native Java实现的zlib压缩可以和C++实现相比较，但如果尝试不同的压缩技术（例如Bentley-McIlroy long common strings），你就需要用Java实现它们，这会产生较差的性能，或者你可以选择用C++实现，通过JNI调用。不过JNI调用的方式丢掉了Java带来的所有好处，而且需要通过JNI调

用方法带来的额外开销。

Boost是一个准标准库，相当于STL的延续和扩充，它的设计理念和STL比较接近，都是利用泛型让复用达到最大化，它并没有修改C++的任何规则。对比STL，Boost更加实用。STL集中在算法部分，而Boost包含了不少工具类，可以完成比较具体的工作。使用Boost，将大大增强C++的功能和表现力。Hypertable选择使用Boost开发，大大增强了代码的可读性、可维护性及开发效率。

总体架构

图1是Hypertable的总体架构示意图可见，Hypertable主要由Hyperspace、Master、RangeServer和DFS Broker等组件构成。

Hyperspace：用来存储一些元数据信息，同时提供一些锁的功能。BigTable论文中，这个功能模块叫Chubby，类似的开源实现是Zookeeper。在Hypertable当前版本中，Hyperspace是个单机系统，基于Berkeley DB实现，由于单点实现，Hyperspace不能失效，否则整个系统会故障。不过在未来，Hyperspace可能基于类似Chubby的集群实现。

Master：和Client交互，接受客户端命令，进行元数据操作；管理RangeServer，检测RangeServer的状态，实施负载均衡策略；定期扫描元数据做GC。

RangeServer：管理多个Range，提供服务，每个RangeServer 管理一张表的部分数据（子表Tablet）。RangeServer是真正操作的执行者，包括读、写等，同时CellCache（内存）和CellStore（磁盘）的组织结构也由它维护。

DFS Broker：主要作用是使用底层的文件系统来完成Hypertable对文件系统的请求。Hypertable 对文件系统的使用有一个很简单的接口，只需要文件系统提供几个很简单的操作就可以。Hypertable的读写文件等操作都是以Socker形式向 DFS Broker发出请求来完成。对于不同的文件系统，只要实现一个 DFS Broker就可以与Hypertable一起工作了。

文件系统：Hypertable设计是为了使用分布式存储系统，当然也可以使用本地文件系统。目前较多使用的是HDFS。

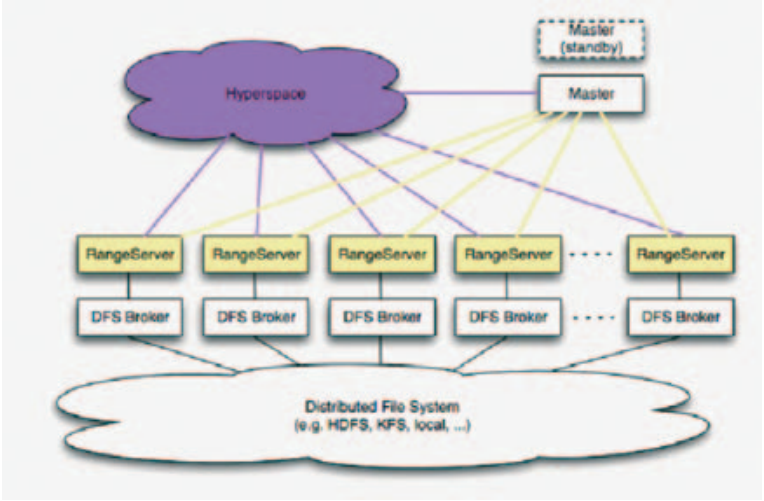


图1 ypertable总体架构示意图

性能评测（Hypertable vs. HBase）

提到Hypertable，就不得不说HBase，这两个项目的出发点是相同的，即BigTable的开源实现。Hypertable是C++语言编写的，而HBase是用Java语言编写的。与Hypertable相比，HBase参与开放开发的时间更长，提交者及外部捐助者的数量更多；而Hypertable的优势在于其高性能。

下面引用Hypertable官网上列举的一些测试来对比两者的性能，对比双方分别是Hypertable 0.9.3.3和HBase 0.20.4。测试将按照BigTable论文中第7节所涉及的测试模型，此外还包括以下改动：系统都运行在10个节点的HDFS（0.20.2版本）集群上；Tablet Server使用5GB的RAM；为了最大化吞吐，每台客户端机器可能运行6个或8

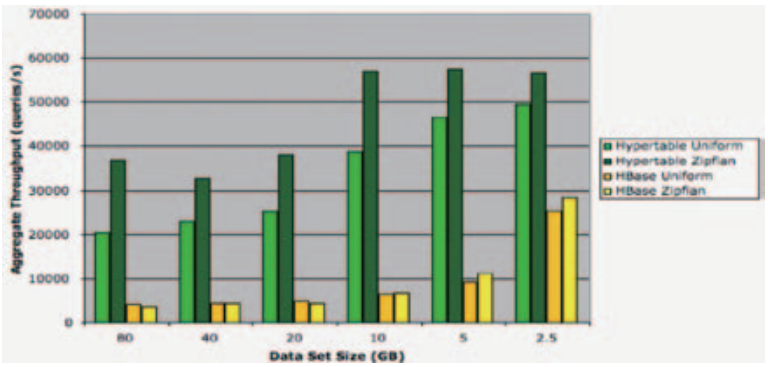


图2 Hypertable vs. HBase 随机读吞吐量测试

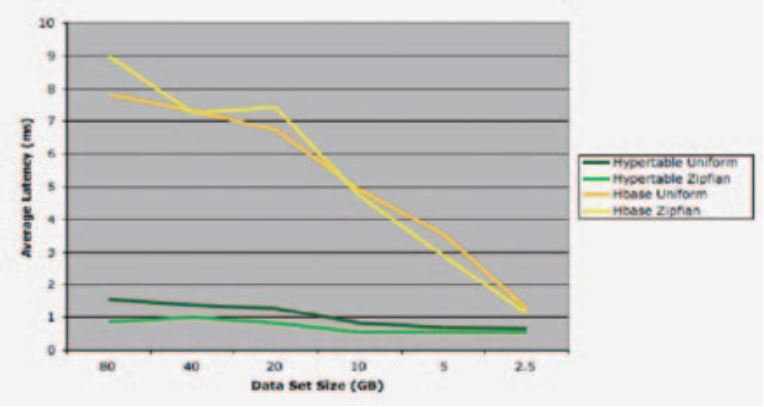


图3 Hypertable vs. HBase 随机读吞吐量测试

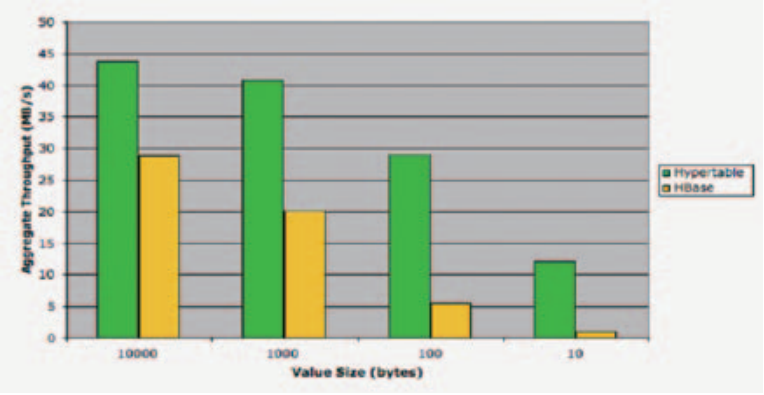


图4 Hypertable vs. HBase随机写吞吐量测试

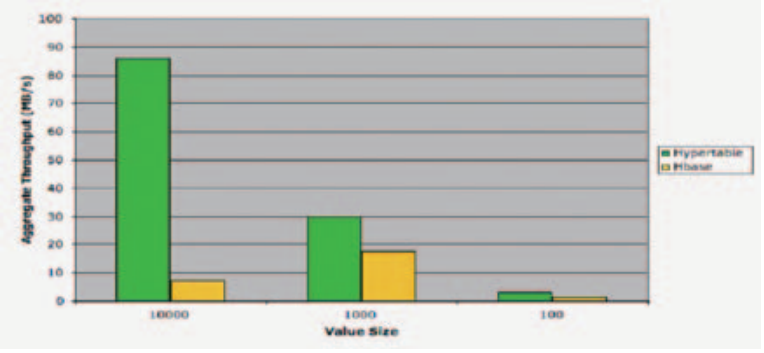


图5 Hypertable vs. HBase 顺序读吞吐量测试

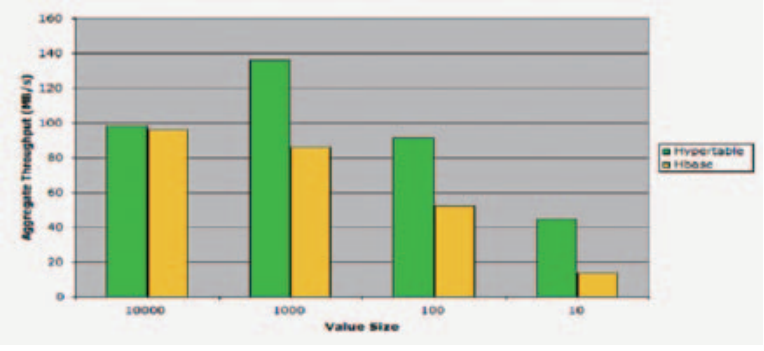


图6 Hypertable vs. HBase Scan吞吐量测试

个Client。

■ 随机读吞吐

数据的Key按照均匀和齐夫两种分布，数据的Value大小固定在1KB左右。数据集的大小在80GB、40GB、20GB、10GB、5和2.5GB间变化，用来测试不同大小的数据从内存到磁盘时，系统性能的变化。每个机器8个Client，总共32个Client，发送2千万个Queries。

图2的测试结果显示Hypertable的性能超过HBase数倍，这是因为Hypertable动态调整分配给每个子系统的内存，依据当时的负载状况。对于读密集型的负载，Hypertable分配大部分内存给Block Cache；而HBase则是固定分配20%的Java Heap作为Block Cache。当数据集是2.5GB时，HBase和Hypertable的性能差距最小，这是因为数据集的大部分都能给被HBase的Block Cache所容纳。齐夫分布时，Hypertable的性能表现更好，这是得益于它的Query Cache。

■ 随机读延迟

图3是图2的转化视图，延迟结果是通过累加每个客户端的所有用时，再除以发出的Queries数量。

■ 随机写

测试随机写，插入80GB的数据，每次测试写入的Value大小从10KB到10Bytes。Key按照均匀分布随机产生。每个机器运行6个Client，总共24个Client。两个系统的写操作都使用了12MB的写缓存。

■ 顺序读

顺序读测试使用随机写插入的80GB数据，每个机器运行8个Client，总共32个Client。顺序读Row Key，每个读作为一个独立的请求。对于两个系统，读返回的Cell的版本数设为1，这会使得读到的数据比插入的要少。因为数据随机插入，某些行是空的，而某些行插入了多个Cell版本。

如图5所示，HBase在处理Value大小为10KB的数据时性能极差，这是多次运行后的结果，目前未定位HBase对大Value处理的问题所在。

■ Scan

Scan测试使用随机写插入的80GB数据，每个机器运行8个Client，总共32个Client。通过两个系统的Scan接口读Row Key。

当Value大小为10KB时，HBase的性能甚至超过了Hypertable。实际上，这是因为HBase将最近插入的数据都缓存在了内存的Memtable中；Hypertable为了最小化故障恢复不可用时间，强行对Commit Log做了Compaction。Hypertable中有“Shadow Cache”的选项，默认是关闭的；打开该选项，最近Compacted的CellCache（Memtable）将继续留在内存，映射磁盘上的CellCache（SSTable）。当打开这个选项后，10000Bytes、1000Bytes、100Bytes和10Byte的测试结果分别是134MB/s、151MB/s、112MB/s和46MB/s。

小结

Hypertable和HBase是一对孪生兄弟，都是BigTable的开源实现。

Hypertable的优势在于基于Boost C++库的高性能实现，使其在性能这个层面超越了HBase，对于那些对性能非常看重的应用，Hypertable是更好的选择。当然，Hypertable的劣势也很明显，就是在开源社区的活跃度不够，贡献力量的人不多，从某种意义上讲，Hypertable是天才Doug Judd的个人作品秀。

HBase的优势恰恰是Hypertable的劣势，作为Apache的顶级项目，HBase在业界的影响力和使用的广泛度要远超Hypertable，像Facebook、Yahoo！、淘宝等公司都在大力推广使用HBase。众多的参与者推进了HBase的成长，其在项目成熟度上目前要领先于Hypertable。此外，使用Java的开发者数量要超过使用Boost的开发者，这也使得很多公司更乐意选择HBase，以便后期更好地维护。

当然，Hypertable也在快速发展，预计2011年下半年会发布1.0版本，实现BigTable所涉及的所有功能，相信到时候Hypertable将会吸引更多的公司和开发者。



郑付联

华东师范大学软件工程硕士，枣庄职业学院计算机系副教授。研究方向为软件工程，先后在国家级刊物上发表过25篇论文。

责任编辑：董世晓（dongsx@csdn.net）

厂商资讯

Android 4.0在香港发布

10月19日，Google与三星在香港联合举办发布会，对外正式发布代号为“冰淇淋三明治”（Ice Cream Sandwich）的Android 4.0系统，以及搭载了Android 4.0系统的新机Galaxy Nexus。

2011透明计算与云计算技术研讨会召开

此次大会由清华大学和英特尔联合举办。中国工程院院士张尧学、清华大学副校长邱勇、英特尔软件与服务事业部中国区总经理何京翔等来自产学研各个领域的专家深入交流了透明计算与云计算技术，共同探讨该领域未来发展方向以及产学研用的结合路径。

第六届中国Linux内核开发者大会召开

10月16日，大会在南京大学召开。大会以“自由、协作、创造”为理念，大力支持开源社区的建设和发展，推动国内Linux内核开发者对国际开源项目的关注和参与。RedHat资深专家Daniel Veillard、《深入分析Linux内核源代码》译者陈莉君、中国资深Linux内核开发者Fengguang Wu等Linux内核国际知名专家做了分享。

第九届Esri中国用户大会召开

本届大会以“GIS——理解我们的世界”为主题，探讨GIS技术发展方向，分享GIS应用经验。据悉，Esri已将公有云GIS、私有云GIS以及面向企业和组织机构的云GIS解决方案云+端落地，而且正在与多个国家级超级计算中心合作，进行产品级云GIS部署测试。

SAP HANA在农夫山泉上线

9月22日，SAP和农夫山泉共同宣布，SAP HANA数据库解决方案在农夫山泉成功上线。这是中国企业首次采用SAP HANA内存计算解决方案，也是SAP亚太及日本区的首个HANA成功案例。作为一种海量数据实时分析工具，HANA能够在不改变现有IT信息系统架构的前提下，实时分析来自几乎任何数据源的海量数据，提高计算速度，从而降低总体拥有成本，大幅提升用户应对市场变化的能力。

达内完成第三轮融资

国庆前夕，高盛注资达内2000万美元，这是达内的第三轮融资。该融资的完成无异于为状态低迷的IT职业教育注入一针“强心剂”。之后，达内又启动了“阳光计划”，将建立移动互联网、云计算、物联网三大人才培养模式研发中心。

微软与中标开展云计算解决方案合作

微软与中标软件于日前签署协议，双方将联合开发、推广并销售云计算解决方案。这些解决方案将基于微软Hyper-V云和中标麒麟服务器操作系统。此外，两家公司承诺在北京成立一个虚拟技术联合实验室，用来开发解决方案，并测试那些让客户迁移到虚拟化和基于云的IT基础设施的云解决方案。

一脉相承的高可用技术

从分布式系统到流式计算

文 / 杨栋

数据流处理涉及的关键技术之一就是高可用技术，而技术选择往往决定了处理系统的架构实现。数据流系统的高可用技术源于分布式系统，因此熟悉分布式系统的相关技术是研究数据流系统的前提。

近年来，一种新的数据密集型应用得到了广泛认同，其特征是：数据不宜用持久稳定关系建模，而适宜用瞬态数据流建模。这些应用的实例包括金融服务、网络监控、电信数据管理、Web应用、传感检测等。在这种数据流模型中，单独的数据单元可能是相关的元组（Tuples），例如网络测量、呼叫记录、网页访问等产生的数据。但这些数据以大量、快速、时变的数据流持续到达，由此产生了一些新的研究问题。

虽然目前没有一个类似于Hadoop的集大成者，但也都各具特色，包括工业界开源的Yahoo! S4、Twitter Storm、商用的IBM StreamBase及学术界开源的Borealis。

数据流处理涉及的关键技术之一就是高可用技术，技术选择往往决定了处理系统的架构实现。数据流系统的高可用技术也是源自分布式系统，熟悉分布式系统的相关技术是我们研究数据流系统的前提。本文从布朗大学Jeong-Hyon Hwang的博士论文*Fast and highly-available stream processing*为起点，谈谈分布式系统和数据流系统的高可用技术。

数据库系统的高可用技术

像其他机械或者电子设备一样，计算机系统会发生故障。故障原因有多种，包括磁盘损坏、掉电或者软件错误等。因此，数据库系统需要一种恢复机制，能够检测故障并引导系统回到故障发生前的状态。早在20世纪70年代，数据库系统故障恢复方面的研究就开始了。先来回顾一下传统的基于磁盘的Rollback技术、Process-Pair方法、持久化队列，以及分布式数据库和工作流系统中使用的高可用技术。

基于磁盘的Rollback

除了硬件故障和软件Bug，数据库系统还可能遭受到事务故障（Transaction Failures），原因可能是完整性约束的违背和死锁。如果发生这种故障，数据库的状态可能会变得不一致。因此，我们需要恢复机制能够保证事务的原子性，这通常分为以下几种。

- Shadow Paging

Shadow Paging使用两个页表，影子页表保存事务开始时的状态，当前页表保存事务操作时的状态。前者在事务销毁时变成新的页表，后者在事务提交后变成新的页表。

- Log-based Schemes

基于日志的策略中，所有更新操作记录日志到持久化存储，如果发生故障，系统通过重放这些日志，使得存储系统恢复系统到先前的一致状态。

通过定期产生Checkpoint，系统能够减少恢复时需要处理的日志记录数量。Checkpoint在持久化存储中保存了一致性状态。基于日志的恢复技术广泛应用于数据库产品，包括IBM DB2和Microsoft SQL Server等。

这些数据库管理系统的恢复技术不是很适合于数据流处理。原因在于存储数据到磁盘的方式会带来很大开销，而且数据库事务的理论也不是适合于持续的数据流处理模型。

Process-Pair

数据库系统常常通过从源数据库复制数据来防止系统故障时造成的数据损坏，源数据库称为Primary，目标数据库称为Standby。这个模型常常称为Process-Pair模型。Process-Pair有几种变化方式，区别在于运行时开销和恢复速度。

在冷备模式下，Primary周期性地传输操作日志到Standby，基于这些日志，Standby异步执行这些操作。尽管操作日志常常被用到，但Primary也可以通过定期Checkpoint自身状态，以便减少恢复时需要处理的日志数量。

在热备模式下，Primary和Standby同步执行所有的操作（例如，在发送结果到客户端之前执行每个更新操作）。相比执行性能，热备模式更重视恢复速度。冷备模式有相反的特性。

Process-Pair模型广泛应用于许多已有的数据库系统，包括IBM DB2 HADR 、Oracle 10g/DataGuard和MS SQL 2005's Database Mirroring。Oracle 10g/DataGuard基于Oracle Streams构建，DataGuard支持三种恢复模式：maximum protection（MPR）、maximum availability（MAV）和maximum performance（MPE）。MPR同步更新到作为相同事务部分的多个机器，提供精确恢复。MPE异步传输重放日志到Standby，仅仅提供间隔恢复（Gap Recovery）。基于Standby的可存取性，MAV在MPR和MPE间转化。

数据流处理中，考虑到应用需求，常常将Process-Pair模型加以变化，称为Passive Standby、Active Standby和Upstream Backup。这些技术能够提供精确恢复或者无损恢复。与精确恢复相比，无损恢复能够提供较少的运行时开销和更快的恢复速度。根据恢复速度和资源使用，这三种改进的恢复技术拥有各自不同的特点。

持久化队列

Bernstein等人为客户端和服务器之间的事务请求开发了一种容错机制。利用持久化队列（Persistent Queues），它们的方法保证每个服务器只处理每个请求一次，保证一个客户端至少处理每个应答一次。

这个方法也不适合于数据流处理，因为它依赖事务的语义，并且在得到处理之前存储消息到持久化存储。数据流处理中有类似的方法，存储数据流源头的输入Tuples。不像持久化队列，数据流处理中保存Tuples在本地内存，直到下游算子处理完成。

分布式数据库的HA技术

分布式数据库包括一系列站点，每个站点维护一个本地数据库系统，每个站点也会参与全局事务（例如，访问多个站点数据的事务）的执行。为了在服务器和网络故障时也能保证全局事务的原子性，分布式数据库通常使用两阶段提交协议（Two-phase Commit Protocol）。

当所有全局事务相关的站点都通知事务完成时，协议才被调用。两阶段协议是投票阶段和决策阶段，前者事务协调者检查是否每个相关站点都准备提交，后者协调者通知所有站点它对全局事务的决定（Commit或Abort）。两阶段提交协议也许会导致一种状态，事务的结果不能确定直到某个站点恢复。三阶段提交协议能够减少阻塞的可能性。

尽管Process-Pair方法能够保护系统免受机器故障的影响，分布式数据库需要处理网络故障和大规模机器故障。因此，复制数据库到不同远程位置的技术被提出。这些副本技术主要分为Eager Replication和Lazy Replication两类。

Eager Replication保持所有副本同步更新，这些副本作为原子事务的一部分，该方法常常导致序列化的执行，因此很难并发；另外，Eager Replication增加了事务的响应时间，因为对每个事务增加了格外的更新和消息传递。

对比Eager Replication，Lazy Replication在可用性和一致性间取得平衡。事务发生时，只有一个副本更新，其他副本异步更新，相当于每个节点的独立事务。如果副本有不一致的状态，系统负责协调状态。

这些副本技术在广域网范畴的数据流处理中是无用的，因为数据库事务的理论无法应用，由于高速的输入速率，算子副本的状态改变得太频繁。因为这个原因，广域网中的数据流处理，算子副本并发执行要比算子副本之间传递更新更加有效。

工作流系统的HA技术

高可用技术在工作流管理系统（WFMS）的上下文中进行研究。在工作流系统中，例如IBM WebSphere MQ，数据传输通过几个达到共同目标的执行步骤。这些工作流系统通常使

用Standby服务器来屏蔽服务器故障。许多解决方案提交每个执行步骤的结果，同时存储消息到持久化存储，能够被Primary和Standby都访问到。为了达到可扩展方式的容错，一些工作流系统记录消息日志到处理节点而不是中心存储。变化的Process-Pair方法在Exotica Workflow System中使用，没有Checkpoint处理状态，Exotica记录工作流组件之间的消息日志。这个方法类似于数据流处理中的Upstream Backup，系统状态能够通过重放日志来恢复。

通常的，工作流系统的HA技术不会直接应用到数据流处理，因为它们依赖事务语义，这通常不适用于数据流处理的上下文。

分布式系统的高可用技术

Rollback Recovery

这类技术将分布式系统视为一系列应用过程，它们通过网络进行交互。这些技术周期性地Checkpoint应用过程的状态到持久化存储。当发生服务器故障时，应用过程从保存的状态重新开始，因此减少了丢失计算的数量。这些Rollback Recovery技术形式产生了一个的全局（系统范畴）一致性状态，作为恢复结果，过程的状态反映了消息的接收，发送者的状态反映了消息的发送。早期的Rollback Recovery技术通过使用Checkpoint来实现容错。这些基于Checkpoint的技术分类如下。

- Uncoordinated Checkpointing

该方法允许每个过程独立执行Checkpoint。发生故障时，回滚过程到早期的Checkpoint，直到全局性的一致性状态形成。这种串联回滚是Uncoordinated Checkpointing的主要缺点，一个过程或许拿到无效的Checkpoint，而它并不是全局一致性状态的一部分。

- Coordinated Checkpointing

过程协调它们的Checkpoint为了保存一个全局一致性状态。Coordinated Checkpointing简化了恢复（由于过程仅仅需要从它们最近的Checkpoint重启），并且减少了存储开销。不过这个方法会导致较大的延迟，因为在任何消息被发送到外部之前，全局Checkpoint必须执行。这类技术保证了全局一致性，通过在发送任何应用消息之前，周期性地让所有过程执行Checkpoint。

- Communication-induced Checkpointing

基于每个应用消息上的协议信息，接收过程能够决定是否应该强制执行一个Checkpoint，以防止上个Checkpoint无效。除了这些强制的Checkpoint，每个过程能够自动产生本地的Checkpoint（例如，当状态很小且因此产生Checkpoint开销）。因为这种方式不需要全局协调，所以它能够随着过程的数量增长进行扩展。

基于日志的Rollback Recovery会合并Checkpointing和非确定性的事件的日志（例如，消息的接收和硬件中断）。理论上，基于日志的Rollback Recovery依赖分段式的确定性，假设一个过程处理的所有非确定性事件能够被标识，并且在恢复过程中按照精确的原始顺序重放这些事件，过程就能够重建故障前的状态。因为这种方法能够恢复系统到某个故障点（而非最近一致性Checkpoints集合），对于那种需要频繁与外部世界交互的应用，这种方法是非常有吸引力的。基于日志的Rollback Recovery有3个变化形式。

- Pessimistic Logging

当事件的效果能够被其他过程或外部系统可见之前，保存每个非确定性事件到持久化存储。由于这个限制，Pessimistic Logging使得非故障期的性能下降。为了减少开销，一些协议依赖专门的硬件或者记录每条消息到发送者的非持久化内存。

- Optimistic Logging

该方法异步地将非确定性事件存储到持久化存储（例如，首先保存这些事件到内存，然后周期性地刷写到持久化存储）。发生故障时，这种方法可能会产生孤儿过程（状态依赖于尚未记录日志的非确定性事件）。Optimistic Logging技术回滚这些孤儿过程直到没有这种过程残留。在无故障执行时，它们跟踪过程间的因果依赖。因果依赖也被用于记录事件日志，在事件的效果被外部系统看到之前。

- Causal Logging

该方法比Optimistic Logging有更好的无故障运行性能，并且保持了Pessimistic Logging大部分的恢复优势。Causal Logging保证，每个

有原因地影响过程的非确定性事件，都总会被写到持久化存储或过程的日志中。为了实现这点，过程基于应用消息上的因果信息，并基于这些消息更新它们的日志。

在数据流处理中，数据流由各个算子连接起来，代表算子之间的依赖。因此无需任何复杂机制，相互独立的算子就可以轻易找到。利用这个特性，技术上可以产生细粒度的Checkpointing，从而能大大减少Checkpoint阻塞规则处理的持续时间；而且，可以做到分布式的备份框架，使得恢复可以并行；另外，备份的分布式部署及按策略调度Checkpoint能够提升恢复速度。

Group Communication

在分布式系统中，一组远程过程之间的交互需要保证可靠性。例如，应用要求每条消息被传递给所有过程组内成员或者都不传递；并且，应用一般要求所有消息按照相同顺序传递给所有进程。以上两个条件定义了广播的原子性。原子性广播在分布式数据库中广泛使用，因为其保证了所有站点按照统一方式更新它们的本地数据库，即使站点或网络发生故障。Paxos算法可以在容错方式下达到分布式一致。这个算法有助于基于恢复技术进行故障恢复，因为它可以让一组副本很容易地对每个成员的健康状态达到一致。

数据流处理的Active Standby模式下，只要多个上游副本有数据到达，下游副本就可以进行处理。这个技术保护系统免于故障影响，即使没有故障监测和副本切换。

数据流系统的高可用技术

大部分数据流处理的高可用技术基于故障恢复。在这些技术中，如果服务器故障，一组预先定义的备份机器将接管失败的执行。通常基于故障恢复的高可用方法有三类。

- Passive Standby

在Passive Standby方式下，每个Primary定期做Checkpoint（即拷贝从上个Checkpoint到当前状态的改变）到它的Backup。

Passive Standby方式广泛用于局域网集群，它能够经受高负载情况，却降低了恢复速度。一些已有技术将每个服务器上的计算进行拆分，将拆分后的部分备份到多个服务器。基于这种分布式备份，能够实现恢复的并行化，因而大大减少了恢复时间。

- Active Standby

在Active Standby方式下，每个Backup也会接收并处理输出数据，和Primary并行，在广域网应用环境下，该方法是个好的选择。

与Passive Standby相比，Active Standby方式保证了更快的恢复速度，低负载情况下所有Backup使用和Primary相同的资源。通过使用分配给Tuples的序列数，Flux技术实现了无损和副本自由的故障恢复语义。Flux允许副本之间松耦合，一个副本能够比其他执行得更快，导致在很多时候执行交迭。Balazinska等人研究Active Standby方法的变体，能够处理网络故障和分割。

- Upstream Backup

在Upstream Backup方式下，每个Primary记录它的输出数据到日志，这样即使下游Primary发生故障，Backup也能通过重放日志数据来从头重新构建丢失的状态。

Upstream Backup在无故障期间执行效率很高，因为其Backup一致保持空闲状态。然而，它可能会花费较长时间来恢复状态较大的计算。例如，恢复一个带有30分钟窗口的Aggregate算子，需要重新处理30分钟内的所有Tuples。对于那些算子的状态很小，并且资源紧缺、故障发生稀少的情况，Upstream Backup是个好的选择。

小结

从以上高可用技术的介绍可以看出，数据流处理的高可用技术基于分布式处理方面的研究，并针对应用特性做了相应的改进。流式计算和分布式计算本身就是一家，更好地了解分布式理论是设计优秀流式计算系统的必要条件。



杨栋

百度分布式高级研发工程师，从事Hypertable、Hadoop及流式计算的研究和开发。

责任编辑：董世晓（dongsx@csdn.net）

基于Minitransaction构建可扩展分布式系统

文 / 韩永国　徐永睿

高可扩展性的分布式系统几乎都采用了消息传递机制，消息传递虽然避免了共享内存容易产生竞态条件等缺陷，但在设计与实现时，需要处理大量复杂且易错的底层分布式协议。而Sinfonia的最大优势在于将这些复杂协议的设计放到更高的抽象级别中（Minitransaction），开发时只需关心设计相应的数据结构。

当前，构建可扩展性分布式系统主要采用消息传递机制。消息传递机制具备硬件依赖度低，且在组件间采取明确定义的方式进行通信等优势，相对于共享存储器，消息传递机制更易进行系统排错。但另一方面，对于分布式状态的管理（数据一致性、数据复制等）以及系统并发、故障处理，采用消息传递机制的系统必须构造复杂的协议来进行相应处理。作为SOSP 07最佳论文——*Sinfonia: a new paradigm for building scalable distributed systems*，提出了一种基于Minitransaction的构建可扩展分布式系统的范型。本文基于这篇论文，重点介绍Sinfonia所采用的Minitransaction。

Sinfonia系统结构

在Sinfonia中，节点分为两种类型——application node和memory node。application node主要运行不同的应用，而memory node则在内存或外存中存放各种应用所需的数据。

application node运行Sinfonia的user library，然后利用minitransaction去访问memory node之上所需的数据。通过将不同的操作尽可能分布在不同的application node上独立运行，提升系统的可扩展性。

memory node以原始线性字节序列而非类型、记录、表等方式存放数据，避免了不必要的耦合。数据采取{memory-node-id, address}方式进行访问，这种方式提升了数据的节点亲和性，避免了采用全局单一地址空间方案（分布式共享存储器）逻辑上近似数据分布在不同物理节点产生的多次访问的性能开销。

Sinfonia自身也不在application node上进行数据缓存，而是为应用提供支持，由应用本身进行缓存控制。这种设计提升了缓存系统灵活性，简化了Sinfonia本身，并且由于通过Sinfonia访问的数据都是数据的最新版本，简化了开发过程。Sinfonia系统结构如图1所示。

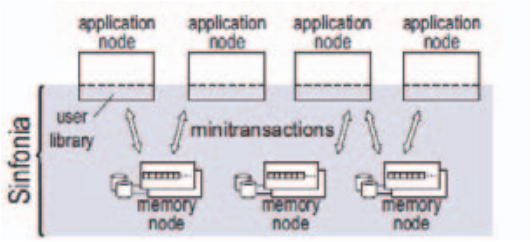


图1　Sinfonia系统结构

从上述介绍可以看出，Sinfonia相对更接近于传统的分布式共享存储器编程范型，但现实世界中的分布式系统，只有非常少的会采用分布式共享存储器模型进行构建，甚至采用此种模型构建分布式系统被认为是一种反模式，其原因如下。

■ 鼓励构建低效系统。在分布式系统中，当采用共享存储器模型时，由于该模型采用全局单一地址空间，在简化开发的同时隐藏了不同节点存储器在物理上分布的特点，使得远程访问过于容易，产生大量代价高昂且非必要的远程存储器访问开销。而在消息传递模型中，

非本地访问必须采用显式send和receive操作，专用的远程访问操作也使得在代码调优过程中能清晰发现代价高昂的数据访问操作。

■ 共享数据访问性能问题。在共享存储器模型中，共享数据的缓存必须实现相应的缓存一致性协议，比如使无效策略。但由于存储器数据访问的粒度（比如页面）与缓存更新粒度（Cache行）不一致，可能导致经典的假共享等问题，造成缓存的乒乓刷新效应从而严重影响性能。即便是彼此独立的数据，在共享存储器模型中也能带来复杂的并发访问问题。

■ 容易产生竞态条件。由于在共享存储器模型中，对于任意时刻，不同节点上不同的程序都可以访问任意的存储器单元，增加了多个节点访问同一存储单元的概率。竞态条件的产生容易引发非常难于发现的隐藏错误。以存储器访问为例，为了减少写入时延，往往对写入操作进行缓冲处理，而不必等待写入完成。这种松弛一致性模型在减少时延的同时违背了顺序一致性语义。在共享存储器模型中，松弛一致性会造成非常难于发现的隐藏错误。这里以并行计算中经典的互斥Dekker算法简化版为例进行讨论，如图2所示。

```
CPU0:                          CPU1:

flag0 = 1;                     flag1 = 1;
if(!flag1)                     if(!flag0)
{                              {
    //临界区代码                    //临界区代码
}                              }
flag0 = 0;                     flag1 = 0;
```

图2 Dekker算法简化版

初始时，flag0 = flag1 = 0，在顺序一致性模型中，当到达临界区代码时，flag0或flag1中最多只有一个flag为0，因此临界区中的实现一定是互斥的。但在松弛一致性模型中，flag0或flag1的置位可能仍在写入缓冲中，因此在读入flag时可能读取的是过时的旧值，导致临界区被破坏。对于该问题，虽然在各种体系结构计算机中都提供了类似CAS（Compare and Swap）的CPU指令进行解决，但在分布式环境中要解决该问题非常复杂。

在Sinfonia中，针对上述问题，主要采用Minitransaction解决。在介绍Minitransaction前，先简单谈谈Sinfonia的设计原则。

Sinfonia设计原则

■ 减少操作间耦合性以提升整体可扩展性。在分布式系统中，组件间过高的耦合必然会影响到系统整体的可扩展性，同时耦合的组件间也很难进行并行处理。目前，一种比较常见的解除耦合的思路就是组件间尽可能采用异步通信方式。在消息传递范型中，Erlang等编程语言中广泛采用的Actors模型受到较多关注。Actors模仿了真实世界中对象的行为，简化了并行程序设计同时具有非常高的可扩展性，但由于Actors是基于消息传递的，并不适合于Sinfonia的编程范型。Sinfonia在减少耦合性，提升可扩展性方面遵循的是“Across minitransactions, spread load; within a minitransaction, focus load”。换言之，在单一的Minitransaction中要尽可能访问较少数量的memory node，而并行执行的Minitransaction彼此间尽可能依赖不同的memory node。

■ 在试图构造可靠的分布式系统前确保系统中每一个组件都是可靠的。由于Sinfonia中每一个组件都进行了可靠性设计，因此当少量组件（如memory node）发生故障时，系统能够持续性工作，而当大量组件发生故障时，系统也能够较快恢复。为此，Sinfonia提供了如RAM、RAM-Replication、Log、Log-Replication、NVRAM、NVRAM-Replication等不同级别的故障容忍策略以适应不同的应用需求。

Minitransaction简介

在Sinfonia中，Minitransaction为操作多个memory node提供了ACID事务语义支持。根据CAP理论，以Dynamo、Cassandra为代表的部分NoSQL系统为了保证可用性及可扩展性，并未采用强一致性而是最终一致性。最终一致性虽然可扩展性较高，但其一致性模型非常弱，即便是非常适合采用NoSQL的大规模Web系统，最终一致性也可能并不适合。例如，假设一个采用了最终一致性更新的照片分享应用，用户

可以设置其分享的所有照片的允许访问列表，当用户访问其他用户分享的照片时，系统将首先检查该用户是否在被访问用户的允许访问列表中。现在假设A用户依次做了如下更新操作：首先，将用户B从其允许访问列表中移除；然后，上传新的照片集。对用户A而言，由于先移除了B才上传新照片集，他确信B一定无法看到新的照片集。然而，系统运行过程中，可能会将用户权限操作应用到副本R1，将照片集添加操作应用到副本R2，虽然R1和R2确保最终是一致的，但不幸的是，用户B恰好在R2与R1最终同步之前查看了A用户分享的照片，而且访问的又恰好是副本R2，A用户上传的新照片集在完全不知情下被泄露！虽然可以采用类似Dynamo中NRW的方法来解决，但提升了系统整体复杂度。因此，一些较新的存储设施如Yahoo!的PNUTS、Google的Megastore和Percolator等都采用了比较强的一致性模型，而Twitter、Facebook相继弃用Cassandra转而使用强一致性的HBase也与此不无关系。

与此同时，Sinfonia的事务模型也并非是传统分布式系统所采用的2PC。在传统的分布式2PC中，全局范围的协调者（coordinator）系统组件必须阻塞式地等待所有的参与者（cohort）在第二阶段成功提交后的确认通知。这种阻塞式设计对于提升系统可扩展性影响非常大，因此，现在一般认为2PC是一种反可伸缩性的反模式，不适用于构建高扩展性系统。此外，传统的分布式事务处理也针对2PC必须阻塞的缺陷提出了改进的3PC，如图3所示。

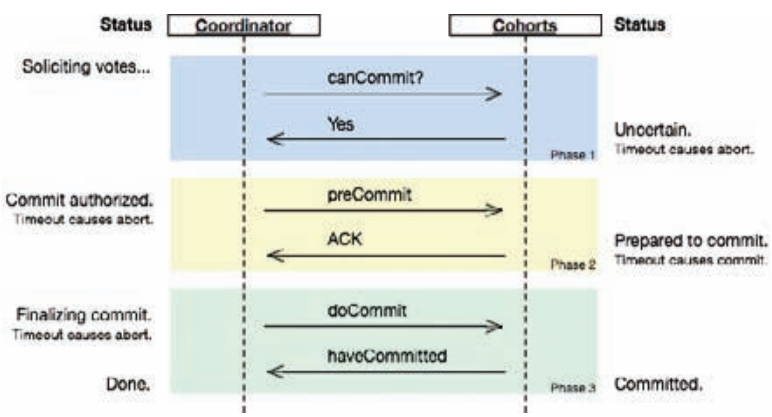


图3 分布式3阶段提交（3PC）算法

3PC与2PC最大区别在于增加了一个PreCommit阶段，只有在指定时间内，所有的参与者都返回了确认消息，系统才会触发commit，同时，3PC的协调者无需采用阻塞方式进行等待。2PC之所以无法采用超时等非阻塞方式，是因为在第2阶段如果协调者和某些参与者同时发生故障，则原有协调者重新恢复或产生新的协调者后必须等待所有的参与者进行响应后才能继续执行操作（假设协调者在2PC的2阶段向部分参与者发送commit通知后发生故障，此时部分参与者可能正在commit，部分已经完成commit，而另外的参与者可能还不知道是commit还是roll back，因为协调者在向他们发出1阶段结果前已经发生故障，而正在或已经完成commit的参与者仍可能在协调者恢复前发生故障）。由图3可知，3PC一旦进入到commit阶段，则所有参与者一定提交，如果此时发生故障，恢复相对容易，而协调者也可以采用简单的非阻塞的超时机制。

3PC采用无阻塞式设计改进了扩展性，但由于需要多次在网络中进行往返，增加了时延，影响了系统性能。在业界中，Google Percolator利用底层的BigTable，通过引入额外的多个列，设计了一种基于2PC但保留了3PC无阻塞式设计的事务提交协议。

在Percolator 2PC的1阶段中，每个事务先在需要修改的一系列Row中确定一个主Row，事务的更新操作都是先对主Row进行更新。当事务开始后，先将事务主Row新值写入data列，并写lock列指向自身，然后依次写入事务所涉及的其他Row的data列，并将这些Row的lock列指向主Row。在Percolator 2PC的2阶段，依然采取先写主Row后写其他相关Row的思想，将write列指向在1阶段中已经完成写入操作的data列，并且清除在一阶段过程中对该记录的lock列写入（清除锁）。等过程全部成功完成后，一个事务才算真正成功提交。在该事务提交之后开始的事务就可以看到最新的数据。

在Percolator事务提交协议中，即便在事务

失败时也不进行回滚（记录上已经加上的锁并不立即释放，同时由于事务并没有提交，即便在1阶段写入了新的data值，也会因为没有2阶段write列的指向，所以后面的事务仍将读取原来最初data值），而是通过后面事务在该记录上检测到锁冲突从而进行Roll-forward和Roll-back。

因为Percolator利用BigTable和GFS保证了服务器端的可靠性，当客户端失效发生在2PC的第1阶段中，后来的事务当访问该事务那些在第1阶段已经上锁的记录时，通过lock列访问到主记录而发现主记录并未提交，因此，简单地撤销该记录的锁就达到了回滚的目的。而如果客户端失效发生在2PC的第2阶段时，后来事务通过该记录的lock列发现主记录已经提交，则认为该事务应该提交，故完成该记录的第2阶段并释放锁（Roll-forward）。

基于如上思路，Percolator无需阻塞，改进了传统2PC的可扩展性，但却导致事务的提交存在着极大的延迟。正如*Large-scale Incremental Processing Using Distributed Transactions and Notifications*中所述：“The design of Percolator was influenced by the requirement to run at massive scales and the lack of a requirement for extremely low latency”。

与上述方法不同的是，Sinfonia中的Minitransaction操作的基本单元是可以根据应用需求进行扩展的一些数据结构，最基本的操作单元是compare item、read item、write item，分别在指定的地址空间进行数据的比较、读取和写入新数据。在图4中，mem-id为memory node编号，addr和len共同决定该节点内部地址区间，而Minitransaction则是以ACID方式操作这一系列的items。

如前文所述，共享存储器系统存储单元一致性问题非常棘手，特别是在分布式系统中。Sinfonia的Minitransaction则非常容易地解决了该问题。下面以一个基于Sinfonia构建的分布式文件系统为例，进一步了解Sinfonia。

在介绍Sinfonia系统结构时提到，Sinfonia本身并不进行缓存，而是交由上层的应用进

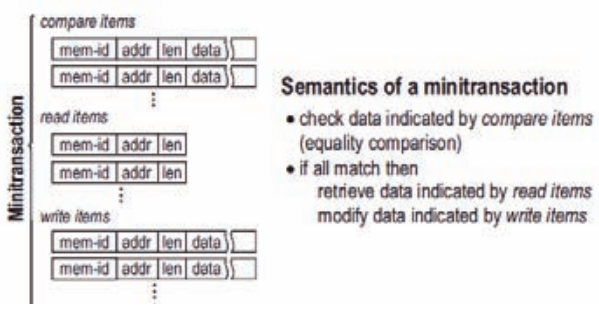


图4 Minitransaction中基本操作单元

```
1 setattr(ino_t inodeNumber, sattr_t newAttributes){
2  do {
3    inode = get(inodeNumber);        // get inode from inode cache
4    newiversion = inode->iversion+1;
5    t = new Minitransaction;
6    t->cmp(MEMNODE(inode), ADDR_IVERSION(inode),
       LEN_IVERSION, &inode->iversion);  // check inode iversion
7    t->write(MEMNODE(inode), ADDR_INODE(inode),
       LEN_INODE, &newAttributes);  // update attributes
8    t->write(MEMNODE(inode), ADDR_IVERSION(inode),
       LEN_IVERSION, &newiversion);  // bump iversion
9    status = t->exec_and_commit();
10   if (status == fail) ... // reload inodeNumber into cache
11 } while (status == fail); }
```

图5 利用Sinfonia的Minitransaction修改inode属性

行控制。在基于Sinfonia构建的分布式文件系统中，运行在Sinfonia application node上的文件系统节点出于性能考虑需要缓存文件系统的inode。而此时，利用Minitransaction的ACID，然后通过使用compare item和write item，就实现了分布式系统中的CAS，保证了一致性（图5中第5~9行）。同时，图5中的setattr方法所对应的getattr方法只需要通过compare item进行inode验证就可以方便地实现分布式环境中的Seqlock。众所周知，Linux 2.6内核引入的Seqlock相对于经典的Reader/Writer，在Reader较多而Writer较少情况下，具有非常高的扩展性，同时也避免了潜在的大量Reader导致Writer饥饿的情形。

Minitransaction协议

Sinfonia中的Minitransaction协议采用的是2PC的改进版，但与传统的2PC相比有如下较大区别。

■ 参与者阻塞而非协调者阻塞。如上文所述，传统的2PC在进行故障恢复时必须阻塞协调者，协调者必须等待所有的参与者发出响

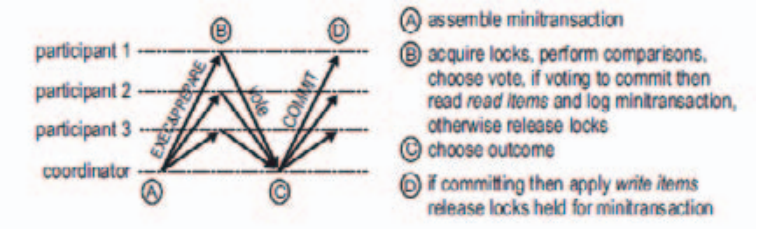


图6 Minitransaction流程

应，而Sinfonia的Minitransaction协议则与之相反。由于协调者运行于application node，参与者运行于相对可靠性较高的memory node，这种故障发生时阻塞参与者的设计事实上降低了系统实际发生故障的概率。其次，memory node能够简单地进行冗余，因此，使用参与者阻塞的策略能较好保障系统可扩展性、可靠性。

■ 协调者不记录任何日志。在传统2PC中，当1阶段所有参与者发回yes响应后，协调者将写入日志标示1阶段完成，同样，在2阶段所有参与者向协调者发回事务成功响应后，协调者也会进行日志记录。而Sinfonia由于协调者无需记录日志，从网络交互来看，由于不需要参与者发回事务完成的响应，直接减少了所有参与者与协调者之间的一次网络交互，同时，也避免了传统2PC在扩展性上最为严重的阻塞问题。Sinfonia中Minitransaction协议流程如图6所示。

■ 参与者尝试获取资源时采用try-lock进行尝试而非阻塞等待。当Minitransaction需要对memory node进行互斥访问时，采用以字为单位的锁粒度在1阶段对数据进行try-lock，如果尝试失败，则返回获取锁失败信息给协调者以便协调者终止当前事务并采用类似TCP中拥塞控制策略在随机指数时间后尝试重试。采用这种方法，避免了分布式2PC中潜在的死锁，但与此同时，正如Sinfonia设计原则中所述，在单一的Minitransaction中要尽可能访问较少数量的memory node，否则大量并发Minitransaction将会造成冲突导致不停进行重试。

总结

当前，高可扩展性的分布式系统几乎都采用消息传递机制，消息传递虽然避免了共享内存容易产生竞态条件等缺陷，但在设计与实现时，需要处理大量复杂且易错的底层分布式协议，如故障恢复、数据一致性等。而Sinfonia最大优势在于将这些复杂协议的设计放到了更高的抽象级别中（Minitransaction），开发时只需关心设计相应数据结构，如write items、read items以及compare items等。与此同时，相对于传统的采用2PC、3PC的分布式事务，Sinfonia中的Minitransaction又能达到非常理想的性能和可扩展性。在*Sinfonia: a new paradigm for building scalable distributed systems*一文中，利用Sinfonia构建了一个分布式文件系统和一个分布式组通信服务原型，原型的实验数据进一步证实了Sinfonia在性能和可扩展性上都有相对理想的表现。

另外，Sinfonia在地址空间抽象处理上级别较低，在开发时需要编写较低层代码进行数据访问。微软的Boxwood【注：参考*Boxwood: Abstractions as the Foundation for Storage Infrastructure*一文】就该问题进行了深入研究。同时，分布式文件系统Ceph【注：参考*Ceph: A Scalable, High-Performance Distributed File System*一文】中采用的抽象层面较高的对象存储思想也提升了Ceph存储灵活性。

虽然Sinfonia目前尚处于原型阶段，但由于其提出了一种新的构建可扩展分布式系统的方法，其中有很多设计值得借鉴。因此，*Sinfonia: a new paradigm for building scalable distributed systems*一文作为操作系统领域顶级会议SOSP 07的最佳论文，非常值得分布式方向和操作系统方向的研究者和实践者阅读研究。



韩永国

西南科技大学计算机学院院长，教授，硕士生导师。中国计算机学会高级会员，四川省计算机学会第五届理事会常务理事，YOCSEF成都主席（2010）。主要研究方向为计算机控制技术、知识管理及其在教育中的应用。



徐永睿

系统架构设计师、系统分析师、软件设计师。现在西南科技大学从事并行计算、分布式存储方面的研究。关注函数式编程语言，经常参与Topcoder算法比赛（Single Round Match）。

责任编辑：董世晓（dongsx@csdn.net）

用云架构解决研发技术瓶颈

记者 / 谭茂

规模化、复杂化是所有研发团队正面临的技术瓶颈，微软亚太研发集团服务器与开发工具事业部中国团队（STB China）用亲身经历告诉我们，如何利用私有云技术整合开发测试资源，以服务的方式提供IT，提升团队协作和开发测试效率，并实现可持续性的绿色IT目标。

2005年初~2011年，随着业务扩展及其范围的增长，STB China的服务器数量递增至千台以上，在业务数量增长和服务器管理任务增加后，依然是由三位工程师组成的研发工程实验室小组负责管理，同时还需要为十几个产品组搭建、维护产品测试环境。和所有业务快速发展的组织一样，STB China正遭遇IT基础设施缺乏战略规划所带来的困扰。

2007年加入STB China的刘擎，目前负责管理团队在上海和北京的研发工程实验室，主要工作内容为向微软的开发、测试和产品经营的团队提供Lab服务，包括测试和搭建环境，以及产品构建和产品设计。

“STB China总共有近1500台服务器，平台基于System Center，总共包括虚拟化服务管理系统、服务器管理、管理客户端、备份管理等四个系统”，刘擎坦言，目前面临运维的压力是巨大的，“STB China在上海的研发工程实验室，连同我在内共有三人，其中一人负责硬件保修，一人负责网络，而且每个人都能成为紧急状况下的全能选手。”

研发正面临技术瓶颈

刘擎认为，要想提升研发团队的工作效率，需要首先了解问题的瓶颈在哪儿。从2010年接手工作之后，刘擎花半年时间在调研内部的流程、资源以及内部工程师的工作习惯。因此在设计私有云时，目标就是解决最优先的三个问题。

“我们需要评估私有云如何提高生产利用率”，刘擎谈道，“对于研发而言，尤其是团队与团队之间有很多重复劳动，比如在做测试时，需要先把Windows部署到一个环境中，然后再构建一个网络环境，运行还没有正式上市的产品。而在构建环境的过程中，不同的团队经常需要做相同的事情。因此这是一个问题，我们需要资源共享。”

此外，工程师对于部署机器所花的时间也非常有意见。比如安装系统至少需要20分钟，如果再安装所有最新的补丁则至少要40分钟，这样的话，整套完成时间至少需要一个小时。“比如物理机在做这个测试时需要做一个AD，而通过服务器，没有两个半小时是完不成的。一天八个小时，工程师一天最多只能做四套。而现在的虚拟化技术可以并行去部署，每一台虚拟器，从开始部署到结束只需要20分钟左右时间。从两个半小时缩减到30分钟，这是一个质变。”

“此外，研发的特点在于，经常需要服务器异地同时工作，也至少需要两个人工作在同一个平台上，北京或者美国的一些数据中心的服务器需要加到上海，因此我们就需要把它整合到一起。”

用云架构解决瓶颈问题

为了解决研发团队面临的技术难题，刘擎花了近一年时间用412台服务器搭建了一个私有云，总共运行了5000多台虚拟机。而在整个管理平台上，团队的资源就是按照分组的方式，支持近20个项目组，会把他们分配到不同的组里，每个用户会在自己的组里分配虚拟机资源。

刘擎首先谈到了对于虚拟化的理解。他认为，虚拟化有两种工作模式，一个是资源优先，另一个是性能优先，目的在于少占资源多花时间和少花时间多用资源。对用户来说，需要比较有弹性，能处理各种紧急需求，非常灵活。“我们一直在追求资源利用率的最大化，过去每个团队把自己的资源放在一个实验室里面，相互之间没有很好的共享平台，现在有了虚拟化管理，它可以很容易共享团队资源。比如过去有个项目需要上，碰巧别的团队也在用服务器，那么除了购买新的服务器就没有别的选择了。现在有个选择，可以从别的团队调一部分资源过来，因为所有的机器都是虚拟化的，不会相互干扰。”

刘擎进一步谈到了具体的实现。据了解，SCVMM的SDK有一个基于Power shell的自动化脚本。作为运维工程师，首先需要了解需求，把共同的需求抽象出来作为一个服务包，然后通过SCVMM的自动化脚本将设备管理实现自动


刘擎：STB China的目标是1000台服务器只配备一位工程师

化。这样用户用起来就不需要重复自动化的工作，这也解决了团队之间的重复劳动问题。

“原先自动化是基于硬件的，现在变成虚拟化之后，其实自动化还是需要做，但我们把这些写了一个抽象的描述，用户就不需要再使用重复劳动这一层，对他来说就是他想要的，并且我们做了更多的工作，结合了一些自动化脚本。我们可以想象一下，开发者需要一个测试环境，首先需要两个CPU，然后需要几个Windows，再需要这个Windows变成AD，那个Windows变成Exchange。把这些选好之后，点一下，后面我们会自动化地把虚拟机准备好，然后调用你所选择好的。这样的话，将运维实现完全自动化，研发人员就能够聚焦在他自己所关注的技术上。在我们的库里提前准备好1200多个虚拟机模板，任何研发工程师可以通过自动化的机制，根据需要进行选择。”

“几年的时间，服务器的数量增长了2.5倍，但我们研发工程实验室的人数并未增加”，在谈到近几年机房的扩容时，刘擎认为，“强大的管理平台再加上IT流程自动化管理工具Opalis是能达成的关键所在。”

刘擎表示，尽管工作效率已经提升了很多，但STB China的数据中心还并没有达到微软的内部标准，“我们的目标是1000台服务器只配备一位工程师。”

寻找机遇 创造未来

CSDN 热门职位全新推荐

UFIDA 用友

用友软件

诚聘以下职位：

一、Java开发工程师

岗位要求：

1. 两年以上Java开发经验；
2. 熟悉SQL Server或Oracle数据库；
3. 熟悉Eclipse开发工具；
4. 熟练掌握SQL语句，能够熟练进行数据库设计。

二、.NET开发工程师

岗位要求：

1. 两年以上.NET开发经验，具备一定分析设计能力；
2. 良好的C#代码实现能力，熟悉ASP.NET开发技术；
3. 掌握面向对象的设计方法和设计原则。

简历投递：rdzhaopin1@ufida.com.cn

DeNA China

上海纵游网络技术有限公司

我们正在全力招聘中国人才，打造一个全新的中国团队，致力于中国智能手机游戏平台的构筑和运营。愿意加入我们，一起构筑世界著名的智能手机游戏平台吗?

Android/iPhone/Web application，更多职位招聘信息请登录公司主页进行查询

公司主页：www.denachina.com

请以“职位+姓名+CSDN”作为应聘邮件标题，将中英文或中日文简历发至hr@denachina.com



主持人：张银奎
《软件调试》一书作者，从事软件开发和研究10余年，对IA-32架构、操作系统内核、虚拟技术，尤其对软件调试有较深入的研究。翻译（合译）作品包括《数据挖掘原理》、《机器学习》、《人工智能：复杂问题求解的结构和策略》、《观止——微软创建NT和未来的夺命狂奔》等。

漫谈Android系统的调试模型（下）

前两期文章介绍了Android系统支持调试的设备端设施和主机端设施，本期将介绍这些设施如何相互协作以提供各种调试功能。

近日来最大的事件莫过于乔布斯去世，世界各地的媒体都在报道，不管电视、报纸还是网络；不同行业和身份的人都在表达怀念，大到总统，小到平民百姓。打开苹果公司的网站，首页上除了一个小小的菜单条外，便是乔布斯的大幅黑白头像，照片的空白区域写着他的生卒年份1955~2011。点击照片可以看到一段简短的祭文：“一位富有远见、充满创意的天才离开了 Apple。一位杰出的、了不起的人物告别了世界。曾有幸与他结识并共事的我们，从此失去了一位挚友、一位精神导师。史蒂夫留下了一家唯有他才能创建的企业，他的精神将成为苹果永续前进的基石。”

在对乔布斯的众多赞语中，特别打动我的是史玉柱说的两句话：“伟大的乔布斯坚持：战略上藐视困难，不受任何教条约束；战术上重视困难，细节上疯狂追求完美。”被打动的主要原因是深有同感。以分号为界，史玉柱的话包含了两层意思。如果用我的话来表达这两层意思，那么就是特立独行和把简单的事做完美。无论是公司，还是个人，都普遍缺少这两种素质。而伟大的乔布斯拥有并完美地展示了这两种素质。特立独行难吗？难！别人做什么，就跟着做什么，别人怎么做，就跟着那么做，但乔布斯和苹果不是，他们敢为天下先，敢冒天下之大不韪。把简单的事做完美难吗？也难！尤其在今天这个极其浮躁的所谓快餐时代更难。敷衍了事，蒙混过关太容易了；静下心来，踏踏实实做好一件事太难了。

坦率说，使用苹果产品给我最强烈的感受便是“精细”，用大白话来说：“人家这活儿做得真细！”放眼看今天的软件世界，代码堆砌太多，精工细作太少；Bug太多，Debug设施太少。少才显得珍贵，拿Android为例，架构方面有些臃肿，“耗能”较多，但是它的应用程序调试设施设计得非常精细，宏观上有统一规划，微观上精工细作、不厌其烦，可以说Android应用程序的繁荣与如此好的调试设施密不可分，而应用程序的繁荣也正是Android系统成功的一个关键。在前两期的文章中，我们分别介绍了Android系统支持调试的设备端设施和主机端设施，本期将介绍这些设施如何相互协作以提供各种调试功能。

调试新进程归纳

上一期我们介绍了在调试器（Eclipse）中创建一个新的进程并开始调试的过程，回顾一下，主要有以下这些步骤。

第0步：创建ADB进程并建立连接，因为一旦ADT（DDMS）插件初始化后这一步便已经完成了，所以我们称其为第0步。事实上，DDMS会创建一个名为Device Monitor的线程一来监视与ADB的连接，一旦连接中断便会尝试重新连接，如果尝试10次仍没有成功，则会使用如下命令行重新创建ADB进程：

```
0:029> da 000000007f687d0
00000000`07f687d0  "D:\Android\android-sdk\platform-"
00000000`07f687f0  "tools\adb.exe start-server"
```

创建ADB进程后，会再次尝试连接，通常便连接成功了。在任务管理器中强制杀掉ADB进程，便可以触发以上过程，图1显示了Eclipse的控制台窗口中打印出的过程信息。

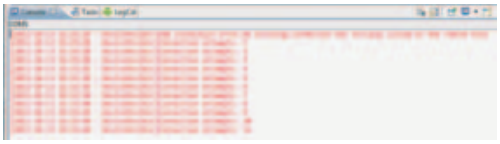


图1 DDMS反复尝试连接ADB

第1步：根据项目属性中配置的发布模式和详细参数选择要运行的目标设备，如果目标设备是虚拟机，那么启动模拟器，包括创建虚拟机进程，运行虚拟机映像等。

第2步：通过ADB将被调试应用程序的APK包下载到设备端并安装。

第3步：通过ADB执行am start命令启动应用程序，在命令行中指定-D开关，这样Dalvik虚拟机创建时就会做好调试准备，创建JDWP服务线程，等待调试器连接，如我们在本文上篇中所讲的那样。

第4步：设备端在成功启动应用程序后会通过ADB向调试器发送一个事件通知，DDMS收到通知后，会通过JavaRuntime.getDefaultVMConnector取得当前调试器（Eclipse）的IVMConnector接口，然后调用这个接口的Connect方法来连接设备端的JDWP服务线程。这样，DDMS便以Eclipse插件的身份“牵线搭桥”，把Eclipse内建的Java调试器与Android设备端的JDWP服务“撮合”到一起了。之后二者就可以通过标准的JDWP协议进行对话了。

图2是来自Eclipse的控制台窗口的日志信息，这些信息反映了在Eclipse上启动应用程序HeiAnd并开始调试的关键步骤。

```
HeiAnd] Android Launch!
HeiAnd] adb is running normally.
HeiAnd] Performing com.HeiAnd.HeiAndActivity activity launch
HeiAnd] Automatic Target Mode: launching new emulator with compatible AVD 'AND233'
HeiAnd] Launching a new emulator with Virtual Device 'AND233'
HeiAnd] New emulator found: emulator-5554
HeiAnd] Waiting for HOME ('android.process.acore') to be launched...
HeiAnd] HOME is up on device 'emulator-5554'
HeiAnd] Uploading HeiAnd.apk onto device 'emulator-5554'
HeiAnd] Installing HeiAnd.apk...
HeiAnd] Success!
HeiAnd] Starting activity com.HeiAnd.HeiAndActivity on device emulator-5554
HeiAnd] ActivityManager: Starting: Intent { act=android.intent.action.MAIN cat=[andr
HeiAnd] Attempting to connect debugger to 'com.HeiAnd' on port 8625
```

图2 开始运行和调试的日志信息

监听对话数据

下面我们通过监听调试器与Davik虚拟机的对话过程来加深大家的理解。我们仍像上期那样，使用jdb附加到Eclipse进程，然后设置如下两个断点，分别监视从Eclipse端发出的数据包和接收到数据包：

```
Stop in org.eclipse.jdi.internal.connect.
PacketReceiveManager.readAvailablePacket
Stop in org.eclipse.jdi.internal.connect.
PacketSendManager.sendAvailablePackets
```

设置好断点后，执行resume命令恢复Eclipse进程运行，按F8，让中断到Eclipse中的HeiAnd小程序恢复执行。顺利的话，上面设置在sendAvailablePackets方法上的断点会命中，执行step命令单步跟踪到SocketConnection的writePacket方法，执行locals命令显示局部变量，然后通过dump命令显示这个变量，其结果如图3所示。

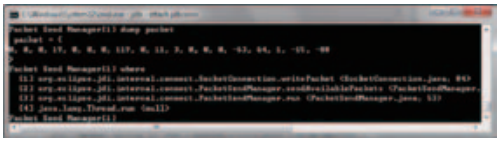


图3 调试器发送命令包

查阅JDWP协议的规约文档（http://download.oracle.com/javase/1,5.0/docs/guide/jpda/jdwp-spec.html），可以知道JDWP包分为命令包和回复包两种类型，两种数据包的开始都是一个固定格式和长度包头，然后是可变长度的数据负载。命令包的包头格式如下：

- 整个包的长度（包括包头），4字节
- 包的ID，4字节
- 标志，目前只使用0x80，代表回复包，1个字节
- 命令集合，1字节
- 命令，1字节

根据这一格式定义，可以解析出图3所要发送包的含义为：

- 整个包共19个字节长，11字节的包头，加8字节的负载
- 包的ID是119，这是一个流水号，用来跟踪包的发送与应答
- 标志为0，说明这是一个命令包
- 命令集合是11，根据协议规约代表的是ThreadReference Command Set (11)
- 命令3，根据规约，代表的是Resume，也就是恢复线程运行

看来这个包与我们按F8恢复HeiAnd程序运行正好对应。执行where观察栈回溯，可以看到当前线程的执行轨迹，线程类为PacketSendManager。

在jdb中执行resume命令恢复Eclipse执行，很快，设置在readAvailablePacket方法上的断点命中，单步跟踪然后使用类似上面的方法可以观察来自Davik端的数据包，篇幅关系不再赘述。

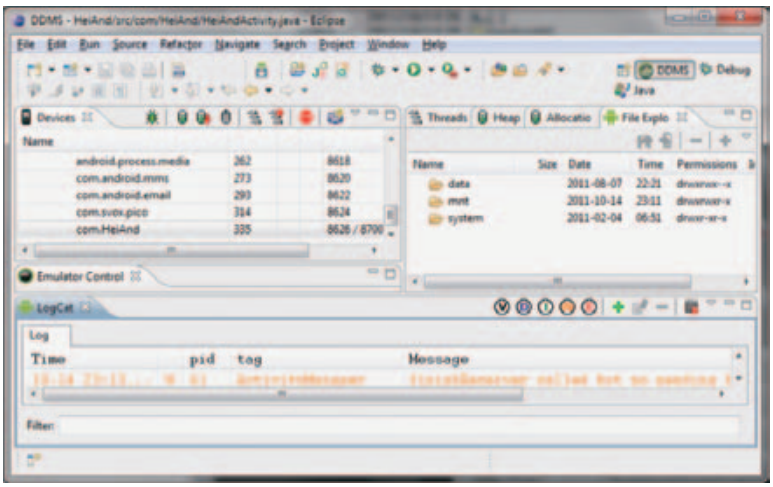


图4 运行在Eclipse中的DDMS界面

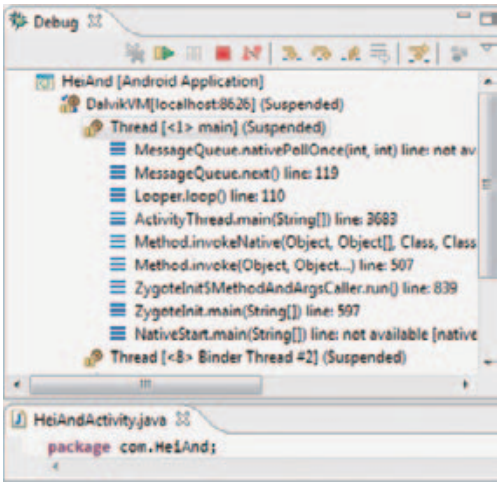


图5 观察主线程的执行轨迹

附加到已经运行的进程

下面再来看另一种情况，那就是将调试器附加到一个已经在运行的Android应用程序。我们先介绍如何操作，然后再简单介绍背后的原理。在Eclipse中打开我们一直使用的HeiAnd项目，点击Run按钮（类似播放按钮，而不是臭虫标记的调试按钮）让这个小程序发布并开始运行，因为没有选择调试，这次这个小程序创建时是没有带调试选项的。

在设备端看到小程序已经运行后，在Eclipse中，点击Window菜单，选择Open Perspective，然后选择DDMS，打开DDMS界面，如图4所示。

图4中有很多窗口，第一次使用可能有些眼花缭乱，我们先将注意力集中在左上方的设备窗口，特别是其中的进程列表、滚动列表，可以看到Android系统中所有应用程序，其中的com.HeiAnd就是我们刚刚启动的小程序。选中这一行，然后点击上方工具条的第一个图标（绿色臭虫）便可以触发附加调试器动作了。如果成功的话，那么这个进程的前面会增加一个同样的图标。

成功附加后，就可以切换到Debug界面使用各种调试功能了，切换的方法有几种，最简单的就是点击图4中右上角的Debug按钮。如果这个按钮被隐藏了，那么可以通过Window菜单选择Open Perspective找到Debug项。

切换到Debug界面后，可以通过Run菜单的Suspend项将被调试程序中断到调试器，中断后可以观察每个线程的当前状态，如图5所示。

接下来可以在根据调试需要在感兴趣的位置设置断点，恢复执行，断点命中后，再观察附近的变量取值等信息。

实际上，附加到已经运行的进程和前面曾详细介绍的创建新进程并调试的工作原理是很类似的，都是通过ADB与设备端Dalvik虚拟机中的JDWP服务器建立对话。使用DDMS的观察线程功能可以看到每个Android应用程序进程都有一个JDWP线程，这个线程就是在等待来自调试器的连接请求。

要使用上述观察线程功能，需要先选择要观察的进程，然后点击工具条中的Update Threads按钮（左数第5个），而后就可以在右侧的线程窗口中看到这个线程的所有线程了，包括管理堆的工作线程和对象回收的GC线程等。双击其中的某个线程，就可以在下方的子窗口中观察这个线程的栈回溯，如图6所示。

观察和追踪堆分配

应用程序所使用的内存主要有两个来源，一个是栈，另一个是堆。而堆是使用动态方式分配内存的主要来源。今天的软件几乎无不严重依赖堆，尤其对于使用Java这类动态语言开发的应用程序，常常有大量的对象实例存储在堆上，而且频繁增加和减少，因此对堆的依赖程度更大。堆使用不当或者堆管理器本身的错误

本期问题：

上期答案是：上一期的问题是Android的模拟器是基于哪个著名的开源虚拟机实现的，答案是QEMU，感兴趣的朋友可以访问http://qemu.org/以了解更多信息或者下载源代码。

本期问题是：DDMS中还有几个辅助调试和支持应用程序调优的功能我们没有介绍，你能说出是什么功能吗？

编者说明：

- 投稿信箱：contest@csdn.net
- 联系方式写在一个单独的TXT文件里，包括以下几项：
 1）姓名
 2）工作单位或学校
 3）电话联系方式
 4）邮寄地址
 5）E-mail地址
- 解答提交时间，最好早于当月15日。

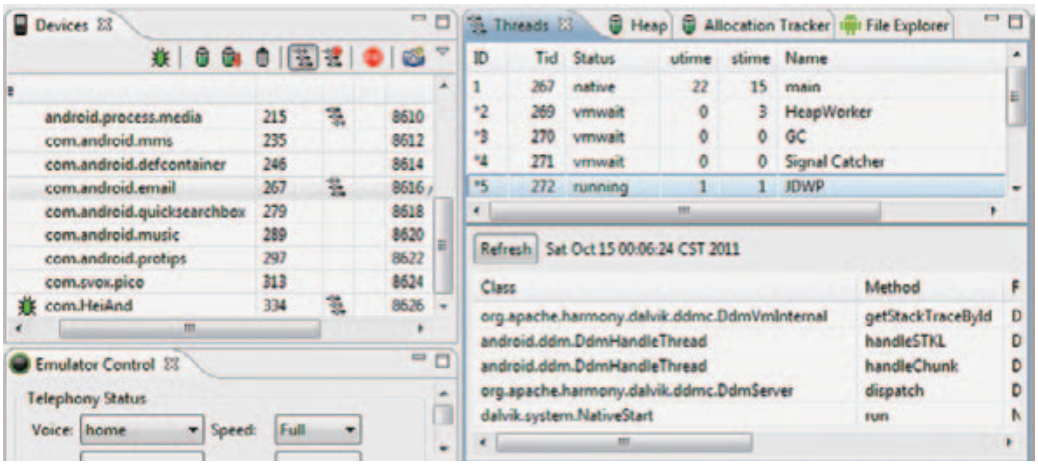


图6 等待调试器的JDWP线程

可能导致随机崩溃等非常严重的软件故障，难以重现和解决。如何调试堆有关的问题是公认的软件难题，即使对于Windows这样的成熟操作系统也是如此。而Android系统所提供的堆调试设施非常完善，强大而且好用，当我第一次见到时甚至感到有些惊讶。

首先堆的调试设施与我们前面所介绍的基本调试设施是水乳交融的，非常好地复用了ADB、DDMS等各种基础设施。从使用角度来看，也是集成到DDMS界面中的，容易看到，而且看了后一目了然。篇幅关系，长话短说，只举两个使用例子。在DDMS中选择要观察的进程，任何应用程序进程都可以，我们就选择com.Android.email程序，然后点击工具条中的Update heap按钮（左数第2个），在右侧选择Heap页，略等片刻，便可以看到所选进程的详细堆信息了，如图7所示。

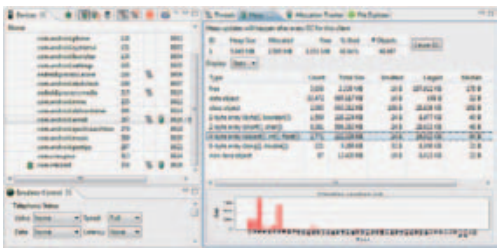


图7 观察堆的使用详情

让我们把注意力集中在右侧窗口，上方的列表列出了进程中的所有堆实例，每一行描述一个实例，目前显示的是1号堆，每一列描述一种重要的属性，包括堆的大小、已经分配空间、空闲空间、分配空间的百分比以及堆上的对象数量。右侧的按钮可以触发内存回收（GC）线程开始扫描堆寻找没有引用的对象。中间的列表是堆中空间的分类统计，比如第一行是已经释放的块，第二行是数据对象，接下来是类实例等。最下方的图标是根据堆块大小产生的直方图统计，横轴是堆块大小，纵轴是堆块计数。

与堆有关的另一个强大功能是分配追踪，如图8所示，上方是堆块列表，选中其中一个，下面便会显示出这个堆块的详细分配过程，也就是调用栈。这个功能与Windows系统中的UST（User-mode Stack Trace）数据库非常类似。

至此，Android系统应用程序调试模型介绍完毕，大家有何感想，欢迎来邮件交流。

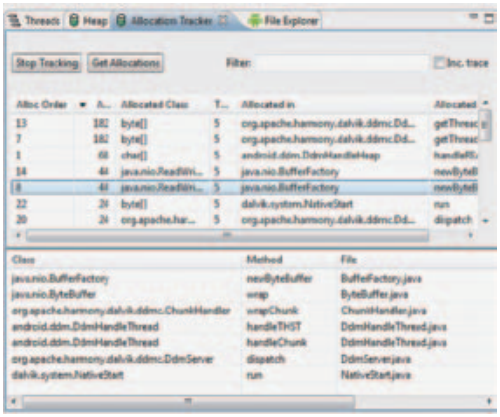


图8 堆块追踪

责任编辑：高松（gaosong@csdn.net）

移动设备的多核时代

文 / 陈冠诚

随着移动设备上多核处理器的普及，越来越多的移动应用将通过并行化的方式提供更绚丽的3D渲染、更流畅的用户体验以及更丰富的特效，这对移动开发者的多线程编程技能提出了新的挑战。移动设备上并行编程时代的大门已经开启。

Apple在今年10月5日正式发布了采用双核A5处理器的iPhone4s，从此宣告iOS和Android两大平台的移动设备全面进入多核时代。

在PC时代，芯片厂商主要通过提升CPU的核心频率，以及升级生产工艺从而在芯片中装入更多晶体管的方式来提升（单核）处理器的性能。然而，从2004年开始，继续提升核心频率和单核的大小会让处理器的功耗呈指数级增加，使得处理器的功耗令人难以接受。而在生产工艺不断进步、单位尺寸能装入更多晶体管的前提下，芯片厂商选择了更加容易实现的多核方案来继续提升硬件的性能。在后PC时代，移动处理器也同样面临着性能的提升与功耗的控制这两大挑战，为了满足提升性能与控制功耗的需求，多核也成为其以后的发展方向。随着NVIDIA在2010年发布了代号为Tegra 2的新一代双核移动处理器，开启了移动设备多核时代的序幕。该款处理器由两个基于ARM Cortex A9的核心及其他视频、音频、图形专用核心组成，是一个典型的异构平台。这个平台架构揭示了移动处理器的两个关键特征：低功耗（比高频单核的处理器耗电量小）以及高性能（异构平台的性能优势）。

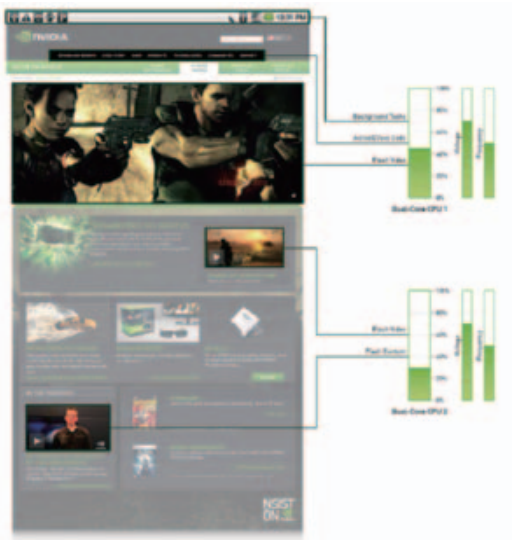


图1　双核CPU使得网页的加载速度更快

对移动设备来说，其常见的应用主要包括浏览网页、播放视频、游戏、短信、电话等。而以iPhone和iPad为代表的后PC时代移动终端更是包括了诸如高清视频摄制与剪辑（1080p）、高分辨率3D游戏（800*400），实时视频通话（Facetime），语音助手（Siri）等对性能要求极高的应用。另一方面，在满足这些应用需求的前提下，我们还需要保持极低的耗电量以保证移动设备能使用足够长的时间。Apple在2008年4月以2.8亿美元收购了PA Semi（Palo Alto Semiconductor）公司，将其150人的芯片设计团队招致麾下以研制高性能低功耗的处理器（iPad和iPhone上的A4处理器以及iPad2和iPhone4s上的A5双核处理器）。事实上，iPad之所以能取得如此大的成功，其十几个小时的续航时间功不可没。而其A4处理器高性能、低功耗的特性对其优秀的续航能力起到了关键的支撑作用。那么，多核到底能给移动设备带来哪些好处呢？

更快的网页加载速度

现在的网页内容越来越丰富，也越来越复杂。HTML5、Flash、JavaScript、视频等内容的

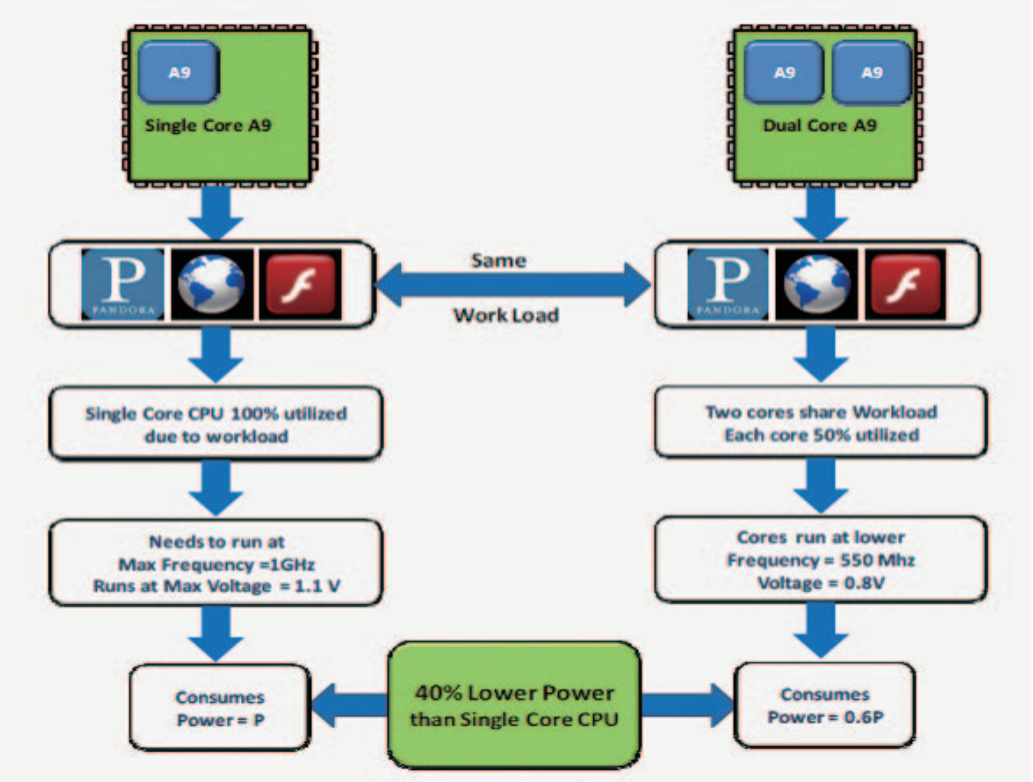


图2　双核移动处理器完成同样的任务只需要相当于单核处理器60%的功耗

呈现都需要强大的处理能力。NVIDIA提供的测试数据表明Tegra 2的JavaScript性能提升了1.5~2倍，网页平均加载速度提升了46%。事实上Firefox、Chrome等桌面浏览器早就已经采用了多线程技术，而Android浏览器和Safari浏览器采用的Webkit内核也已经实现了多线程。在浏览器已经大量并行化的前提下，多核移动处理器自然能提供更快、更丰富的网页渲染体验。在图1中我们可以看到，同一个网页的上下部分可以被分配给两个处理器核心分别进行渲染，这样不仅能有效提升网页的渲染速度，更能保证每个处理器核心不需要跑在最高频率/电压上就能完成任务，有效地保证了较低的功耗。

更低的功耗及更高的性能瓦特比

对多核处理器来讲，任务调度算法及电源管理算法是提升性能瓦特比的关键。Tegra 2能通过如下几点降低功耗：把任务平均分配到两个核心上，这样每个核心都不必跑在最高频率/电压上，而只需要以较低的频率/电压就能完成任务，从而节省功耗；如果要执行的任务是高度并行化的，Tegra 2就能更快的完成这个任务，从而更快地进入超低功耗待机模式，节省耗电量；如果任务只需要一个处理器核心的话，其他的计算单元可以被关闭，从而节省电量。

举例来说，假设在一个单核移动处理器上浏览一个网页，该网页包含了两个Flash视频、图像以及数个脚本文件。那么在大多数时候，这个单核处理器会运行在最高频率/电压上。假设此时的最高频率为1GHz，最高电压为1.1V，其产生的功耗为P。如果我们采用一个双核移动处理器来处理同一个网页的话，该网页的渲染任务会被均匀分配给两个处理核心，每个核心只需要处理一半的任务即可。此时，每个处理核心只需要运行在550MHz、0.8V左右即可以完成任务。如图2所示，最终双核处理器只需要0.6P的功耗就能完成相同的工作。

续航能力一直是智能手机、Pad等移动设备的关键问题，在电池技术没有突破性进展的今天，我们只能寄希望于硬件/软件上的优化来降低功耗，以延长设备的续航时间。

提升游戏体验

Tegra 2的图形处理单元叫做Ultra Low Power（ULP）GeForce GPU，性能较强。如图3所示，现在的一些主流游戏引擎早已经完成了并行化（多个线程分别用来完成渲染、音频、网络、解码、碰撞检测、透明等任务）。相关测试数据表明虚幻3引擎在Tegra2双核处理器上快了将近70%。一个值得注意的地方是，很多游戏引擎是通过任务并行（task parallelism）的方式以适应不同的处理器核心数目，这说明基于这些引擎的游戏可以几乎不用修改程序就能在以后的4核乃至8核移动平台上取得更好的游戏体验。游戏在最受欢迎的移动应用中还是占了相当大比例的，如何使用多线程游戏引擎进行移动设备上的游戏开发将会是技术热点与难点。

Game/Engine	Number of Threads
Unreal Engine 3	4+
Id Tech 5	6+
Frostbite	14
Civilization 5	12
Mafia 2	4
Crysis	8
Uncharted 2	8
Killzone 2	8+

图3　主流游戏引擎使用的线程数

更平滑的用户体验及更快的多任务处理能力

多任务处理在手机/Pad上都非常常见。当你一边听着歌，一边下载电影，一边上网冲浪时，多核处理器就能帮你把这些任务分配到不同的核心上并行处理，从而给你提供更平滑的用户体验。我记得iPad上的一些电子杂志的界面响应速度是个很大的问题，因为它们的渲染速度太慢了。相信当使用性能更高的iPad2时这些程序的界面响应速度能得到较大的提升，当然，其前提是这些电子杂志程序已经多线程化并能充分利用好多核。举例来说，可以用单独的线程来完成类似网络下载这样的耗时任务，同时使用单独的线程来进行用户交互，以提升用户体验。

智能语音助手

我一直认为Siri是iPhone4s中最被低估的新特性。Apple在iPhone4s发布之前的宣传口号是“Let's talk iPhone”，这里面其实就暗喻了智能语音助手这个新功能。我们知道，乔老爷在每一代iPhone中都会引入一些非同寻常的新功能或者新特性，而Siri作为iPhone4s的最大卖点，其作用肯定被很多人低估了。那么Siri到底是什么呢？一个语音识别软件？没那么简单。从用户的角度看，如果想知道诸如最近的天气、附近的饭店等生活信息，你只需要向Siri提问，它就能自动帮你通过服务器连接到Wikipedia、Wolfram Alpha、Yelp等搜索引擎查找相关信息。这意味着Google今天的互联网入口的地位有可能在不久的将来被Siri取代。与此同时，Siri对用户手机上的日历、短信、联系人等私有信息更是有着完全的访问权，这里面的商机不言而喻。如果类似Siri的系统在将来成为个性化移动互联网的入口我一点也不会吃惊。相信不久的将来Google也会强化Android上语音助手的功能。

那么强大的智能语音助手跟多核有什么关系呢？有爱好者已经成功将Siri移植到iPhone4上，但在iPhone4上的Siri运行得非常缓慢，而且还使得iPhone4上的其他应用运行速度也变慢了许多。由此可见，iPhone4s上的双核处理器对提升类似Siri这样应用的运行速度和用户体验有很大的帮助。当然，Siri程序本身肯定已经成功并行化了。

总结

随着移动设备上多核处理器的普及，越来越多的移动应用将通过并行化的方式提供更绚丽的3D渲染、更流畅的用户体验以及更丰富的特效，这对移动开发者的多线程编程技能提出了新的挑战。移动设备上并行编程时代的大门已经开启，移动开发者们，你们准备好了吗？



陈冠诚

IBM中国研究院研究员，瑞典查尔姆斯理工大学（Chalmers）硕士，主要研究方向为并行计算、计算机体系结构及分布式系统。个人博客为并行实验室（http://parallellabs.com）。

责任编辑：高松（gaosong@csdn.net）

基于动态分类策略的ACL控制原理

文 / 汪金保

ACL的重要体现在它决定了系统的灵活度，几乎可称为系统的基石，如何处理好这个问题非常关键。本文作者阐述了基于动态分类策略的ACL控制原理，不失为一种有效的解决方案。

ACL几乎是所有应用系统必须处理的问题，如ERP、CRM、OA等。ACL分为两种层面：

- 功能级权限，判断用户是否有权限使用某项系统功能。如张三是否可以查看订单列表，李四是否可以修改商品价格信息；
- 数据级权限，判断用户是否有权限对某项数据进行特定操作。如张三是否可以查看订单编号为20110523-2333的订单，张三是否可以查看总部订单，李四是否可以修改饮食类商品价格信息。

不少人将ACL看做系统基石。如果不牢靠或处理不当，都会使整个系统失去灵活性。这也正是很多软件企业为某客户开发项目，去另外一家客户实施同样的项目，结果并不是想象中那样做少许修改，几乎是重写整个系统的原因。

权限应用场景分类

从应用场景角度，有如下场景会用到权限：

- 权限菜单树：当前用户有哪些操作权限；
- 查询权限：当前用户能够查看哪些数据，哪些行、哪些列；
- 决策权限：当前用户提交这个数据，允许吗；
- 页面功能级权限要素：当前用户是否具有某项操作，如果有，则显式某项功能按钮、链接等；
- 页面数据级权限要素：比如不同用户下拉框显示不同内容，当前用户查看本照片是否显式删除按钮。



图1 权限应用场景说明

以上情况，可以按照图1做简单对应说明：

- 编号1区域，显示当前登录用户的权限菜单树；
- 编号2区域，显示当前登录用户能够查看到的订单列表（行、列级）；
- 当用户勾选ID=1的订单，点击删除按钮时，后台会做决策权限判断——是否允许当前登录用户删除ID=1的订单（涉及交互，未在图上标示）；
- 编号4区域，显示了“新增”按钮，表示当前登录用户具有新增订单权限；
- 编号5区域，显示当前登录用户机构下拉框可以显示的内容。

权限定义

正确的权限定义应该是这样的：

- 权限定义的主语一定是权限。很多人不自觉的对用户进行权限定义，以用户为主语；
- 每次只针对具体某个权限做定义，不要泛化，覆盖多个权限操作。很多人在一条定义里面既覆盖增加操作、删除操作和修改操作；
- 每个权限按照功能级权限、数据级权限两个层面进行定义。

比如订单管理操作，我们对其定义4个权限：增加订单、修改订单、删除订单和查询订单。分别为这4个权限定义数据级权限；使用RBAC模型来控制4个权限的功能级权限。

RBAC及拓展方案

RBAC（Role Based Access Control，基于角色的访问控制）是目前广泛使用的权限管理解决方案。RBAC使用角色定义对于权限（组）的访问权利，然后将角色赋予用户。判断用户是否有权限访问某项功能时，只需判断该用户具有的角色是否保护该项功能权限。

因此，功能级权限使用RBAC模型非常合适。对于数据级权限显得非常无力。本文推荐使用动态分类策略的ACL方案。

在分析动态分类策略的ACL方案之前，我们先研究很多人采取的方案——基于RBAC的拓展方案。

RBAC扩展方案局限性

不少开发人员对RBAC作扩展，发展成5表模型、8表模型等，对数据级权限进行管理，起到一定效果。这种思路的核心依然是访问控制列表（Access Control List），对数据资源进行静态描述，然后将资源列表赋给用户。这样一来显式对于数据资源使用列表（List ）模式进行权限划分，代价是非常大的：

- 维护工作量大：维护访问控制列表（Access Control List ）工作量非常庞大，属于用户数量(m)×资源数量(n)级别；
- 系统效率低：当用户或者资源发生变动时，都会对访问控制列表进行调整。

其他方案

其他方案主要有：规则引擎；第三方框架，如Spring Security、Shiro。

规则引擎方案比较可行。由于并非专注于权限管理领域（意即缺乏标准），因此随着深入使用，规则引擎不能满足很多权限需求。

框架定义了一套权限编程模式，告知开发者以怎样的模式注入权限判断，在运行时环境，框架会调用相关判断逻辑。框架本身不处理任何权限逻辑，权限逻辑还需要开发者进行编码或者配置。

基于动态分类策略的ACL模型

下面，我将提出一种基于动态分类策略的解决方案。基于动态分类策略的核心思想是：不使用静态的List模式，而使用基于规则描述或者脚本描述进行数据级权限鉴权。

功能级权限依然使用RBAC模型；数据级权限采用动态分类策略的ACL。因此，权限判断顺序应该是：先作功能级权限判断。如果有功能级权限继续作数据级权限判断；否则拒绝用户访问。

数据级权限类型

首先，我们把数据级权限分为2大类型：

- 查询权限，用户从系统获取数据。比如：查询订单数据；
- 决策权限，用户向系统提交数据。比如：修改某条订单信息。

动态分类策略的由来

根据前面“权限定义描述”章节，能够把数据级权限需求描述的非常清晰易懂。

任何查询权限都可以这样描述：

- 什么样的用户，能够执行什么样的数据查询；
- 什么样的用户，能够执行什么样的数据查询。

任何决策权限都可以这样描述：

- 决策（允许|拒绝），什么样的用户，操作什么样的业务数据；
- 决策（允许|拒绝），什么样的用户，操作什么样的业务数据。

以上提到的什么样的用户，我们称为用户分类；什么样的数据，我们称为业务数据分类。都使用规则或者规则组对其进行描述。

这种基于规则或者规则组对用户进行描述，形成对用户的动态分类；这种基于规则或者规则组对业务数据（资源）进行描述，形成对业务数据（资源）的动态分类。

有的时候，权限特别复杂，难以用规则描述，可以考虑使用脚本语言或者原生态编程语言进行描述。我开发的Ralasafe开源中间件支持原生态Java 语言。

规则数据来源和计算方式

以上，我们提到的规则和规则组，会对哪些数据进行计算呢？又能够进行怎样的计算呢？只要规则的数据来源足够丰富，规则计算方式足够有效，那么就能够满足绝大部分对用户、业务数据和数据查询动态描述的需求。

规则读取数据来源应该有：

- 确定的固定值，如：定义总部机构级别=1；
- 用户属性信息，如：读取用户的name信息；
- 业务数据属性信息，如：读取当前借款金额信息；
- 程序上下文信息，有些信息可以放入上下文，权限引擎读取上下文信息，如：读取当前系统日期信息；
- 数据查询，直接对数据库进行相关查询，如：查询当前用户当天借款总额。

规则计算方式有：

- 比较计算（=, >, <, !=, >=, <=），如：判断当前用户的公司级别是否等于1；
- 字符串、对象的equals操作；
- IN计算（IN, NOT IN）；
- NULL比较（NULL, NOT NULL）；
- LIKE操作；
- 表达式组运算，如：多个条件的组合；

■ 非（!）逻辑操作，&&和||逻辑操作。

有了以上计算方式，尤其是表达式组计算方式，就能够满足绝大多数动态描述需求。

查询数据级权限解决方案

基于动态分类策略的查询数据级权限模型如图2：当输入某用户，权限引擎会从第一条策略依次执行；如果当前用户满足某条查询策略的用户分类的规则，则执行该查询策略的数据查询，并返回查询结果。如果当前用户不满足任何用户分类的规则，则说明当前用户没有任何数据级查询权限，返回空集合。

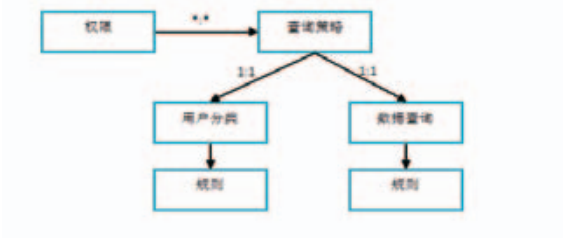


图2 基于动态分类策略的查询数据级权限模型

注意：不同用户对数据库的列级查询权限，在数据查询的select属性体现；不同用户对数据库的行级查询权限，在数据查询的where属性体现；策略的顺序体现优先级：如果某用户同时满足其中多个用户分类，以第一个满足的用户分类为准。

决策数据级权限解决方案

基于动态分类策略的决策数据级权限模型如图3：当输入某用户请求对某数据进行操作，权限引擎会从第一条策略依次执行；如果当前用户满足某条决策策略的用户分类的规则，且满足业务数据分类的规则，则返回该决策策略的决策。如果当前用户不满足任何用户分类和业务数据分类的规则，则说明当前用户对当前数据没有操起权限。返回拒绝。

注意：策略的顺序体现优先级：如果输入用户和数据同时满足多条策略的用户分类和业务数据分类，以第一条满足的用户分类和业务数据分类为准。

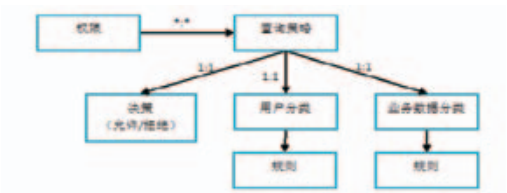


图3 基于动态分类策略的决策数据级权限模型

案例分析

假如有如下查询权限需求：

总部用户能够查询所有员工；

分公司用户能够查询所在分公司及下属营业部员工；

营业部用户能够查询所在营业部员工。

可以使用这样的策略实现查询权限需求：

序号	用户分类	数据查询
1	总公司用户	查询所有员工
2	分公司用户	查询所在分公司及下属营业部员工
3	营业部用户	查询所在营业部员工

注，其中各用户分类的伪代码如下：

总公司用户：`${user.companyLevel}=1;`

分公司用户：`${user.companyLevel}=2;`

营业部公司用户：`${user.companyLevel}=3。`

各数据查询的伪代码如下：

查询所有员工：`select * from employee;`

查询所在分公司及下属营业部员工：

```
select a.* from employee a, company b
where a.companyId=b.id and b.id=${user.
companyId} and b.parentId=${user.
companyId};
```

查询所在营业部员工：`select * from employee where a.companyId=${user.companyId}。`

当总部用户请求查询时，将会命中第一条策略，执行第一条策略的数据查询，返回所有员工；当北京分公司用户请求查询时，将会命中第二条策略，执行第二条策略的数据查询。因为当前用户隶属于北京分公司，返回北京分公司及北京分公司下属营业部员工；当北京海淀区营业部用户请求查询时，将会命中第三条策略，执行第三条策略的数据查询，返回北京海淀区营业部的员工。

一般情况下，营业部用户数量大于分公司用户数量，分公司用户数量大于总部用户数量。所以，第3条策略的命中几率远远高于第2条策略的命中几率，第1条策略的命中几率最低。如果将策略顺序颠倒，会大幅提高策略执

行效率：

序号	用户分类	数据查询
1	营业部公司用户	查询所在营业部员工
2	分公司用户	查询所在分公司及下属营业部员工
3	总部用户	查询所有员工

假如有如下决策权限需求：

ATM取款每次取款额度不能超过5000￥；

ATM取款每天取款总额不能超过20000￥。

可以使用这样的策略实现决策权限需求：

序号	决策	用户分类	业务数据分类
1	拒绝	所有人	单笔借款额大于5000￥
2	允许	所有人	加上本笔借款额，全天借款总额不超过20000￥

注，其中各业务数据分类的伪代码如下：

单笔借款额大于5000￥：`${businessData.money}>5000;`

加上本笔借款额，全天借款总额不超过20000￥：`${businessData.money}+${select sum(money) from loan_money where userId=${user.id} and loan_date=${context.today} }<=20000。`

当张三借款5001￥时，第一条策略被命中，返回决策拒绝；当张三借款5000￥时，第二条策略被命中，返回允许；当张三连续四次借款5000￥后，再借款1￥时，没有策略被命中，返回拒绝。

Ralasafe参考实现

Ralasafe访问控制（权限管理）中间件【注：网址为http://www.ralasafe.cn】，是MIT协议开源免费软件产品。它使用图形化方式处理各种数据级权限管理问题，包括数据库行列级、字段内容级权限问题；它还能处理登录控制、功能级权限控制和URL权限认证。

图3中，权限引擎负责响应API请求，解析相关策略，返回结果；Web管理控制端负责管理、定制和在线测试各种权限策略，不需要编写程序或者手工配置XML。

权限API简介

Ralasafe的权限管理API非常简单，主要是4大API：

- Privilege Ralasafe.getBusinessPrivilegeTree (User user) 获得当前用户的权限树。
- boolean Ralasafe.hasPrivilege(int privilegeId, User user) 判断当前用户是否拥有某功能权限。
- QueryResult Ralasafe.query(int privilegeId, User user) 执行查询权限，返回当前用户执行某项查询时，能查看到哪些数据。
- Decision Ralasafe.permit(int privilegeId, User user, Object businessData, Map context) 执行判断权限，返回当前用户是否可以操作某业务数据。

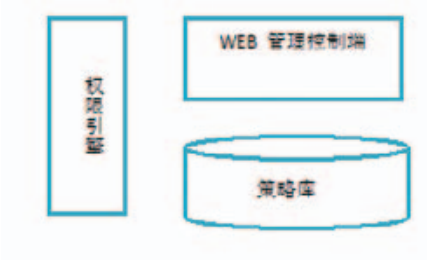


图3 系统架构

推荐实践原则

我们推荐以下实践原则：权限与业务分离；责权分离。

随着对Ralasafe的深入学习，读者会发现能够使用Ralasafe对权限系统本身进行权限控制，实现权限的嵌套管理。

获益分析

总之，使用Ralasafe开源访问控制（数据级权限管理）中间件，开发者主要获益有：

- 权限与业务解耦合——实践更清晰的系统架构；
- 权限管理完全图形化管理，不需要写一行代码或XML文件（基本配置信息除外，如数据源和用户元数据）——实践高效开发；
- 所有访问控制被集中、规范化管理——实践访问模式的统一，减少安全漏洞。



汪金保

Ralasafe开源软件发起人。2004年开始专注权限管理领域，并发起Ralasafe项目。2010年将项目开源。

责任编辑：高松（gaosong@csdn.net）

寻找机遇 创造未来 CSDN 热门职位全新推荐

杭州优朋电子商务有限公司

杭州优朋电子商务有限公司位于风景秀丽的美丽之洲——杭州余杭，是一家高新技术企业，专业从事计算机软件技术研发、信息技术服务及旅游电子商务网络运营。公司独立开发运营的旅游电子商务平台，为国内提供一站式服务的个性化旅游服务平台。

因业务发展需要，急需以下人才加盟，让我们一起来创造美好的明天！

- .NET高级软件开发工程师
- .NET中级软件开发工程师
- Web前端开发工程师
- 网站运营总监

简历投递：zhuqr@uniboncn.com　　联系QQ：5462240

欧特克软件（中国）有限公司

你一展身手的舞台，
打造美好未来

Autodesk

加入我们的队伍，以一种全新的方式实现设计理念。无论是设计可持续性建筑还是开发混合动力汽车，Autodesk软件都可助你一臂之力。作为全球前端的二维与三维设计、工程和娱乐软件提供商，我们致力于帮助打造出更美好的未来。

让我们共同创造未来!

更多职位信息请浏览网站：www.autodesk.com/careers
或直接将简历发送至：acrd.hiring@autodesk.com

方正宽带网络服务股份有限公司

FOUNDER 方正
正 在 你 身 边

公司诚聘：

- 高级研发工程师 (Linux C方向)
- 高级研发工程师(Java方向)
- 高级研发工程师(PHP方向)
- 软件测试工程师

任职要求：

- 本科及以上学历，计算机或网络相关专业
- 熟悉相应的开发语言，有相关开发经验者优先
- 良好的团队合作精神，较强的创新精神

发送简历请以“姓名+职位+CSDN”为标题
邮箱：hr01@founderbn.com
公司网址：http://www.founderbn.com

聚胜万合信息技术（上海）有限公司

聚胜万合MediaV依托强大的互联网广告平台，为广告主、网站主和战略合作伙伴提供专业的互联网营销服务。现热招云计算工程师、PHP工程师、数据处理应用工程师、产品经理等职位，邀您共谋中国数字营销大变局。如果您想在互联网广告领域一展鸿图，如果您拥有足够的经验和才能，MediaV期待您的加入。

MediaV

公司网址：http://www.mediav.com

杭州拓乐网络科技有限公司

诚聘高级网游程序员若干：

- 逻辑程序员
- 客户端程序员
- 服务器程序员
- 网络工程师和计算机管理员

更多岗位信息请访问公司网站： www.tolez.com

简历投递：hr@tolez.com 联系人：徐小姐

厦门梦加网络科技有限公司

客户端程序员 （重点职位） 若干

至少一年相关工作经验，熟悉C/S架构，拥有良好的学习能力和独立解决问题的能力。根据游戏策划和主程序的要求，负责游戏客户端编程和维护工作。

招聘要求：

1. 计算机相关专业，熟练掌握Java或C#编程语言，热爱游戏编程；
2. 使用过Unity3D引擎开发过游戏者优先；
3. 公司统一使用英文版本的技术软件，拥有大量外籍员工，英语口语较好者优先。

公司地址：厦门市思明区厦禾路666号海翼大厦A座11层
公司主页：www.mengjiagames.com
公司电话：0592-2661183骆小姐　　简历投递：hr@mechanist.co

完美世界（北京）网络技术有限公司

完美世界（北京）网络技术有限公司（原完美时空）是中国领先的网络游戏开发商和运营商之一。公司成立于2004年，一直致力于创造优质的互动娱乐产业品牌，倾力打造拥有自主知识产权的高质量网游精品。

现诚聘

- ERP EBS 技术顾问
- flash（as3）开发工程师
- Java开发工程师
- 高级客户端开发工程师
- 游戏运维工程师
- PHP开发工程师
- 游戏服务器端软件工程师
- 游戏客户端软件工程师

职位详情请点击：http://special.zhaopin.com/bj/2011/wanmei03162823/index.htm
公司网址：www.wanmei.com　**简历投递：**zhaopin@wanmei.com

天津港保税区科技发展局

东软（天津）、展讯（天津）高薪诚聘IT英才

招聘职位：高级Java工程师，高级C++工程师，DSP工程师，芯片驱动工程师，应用软件工程师，驱动软件工程师，算法工程师

工作地点：

天津空港经济区（www.tjftz.gov.cn）

简历投递：resume@adm.tjftz.gov.cn

北京法国电信研发中心

Software engineer in the web service system development

Position requirements:

- Master degree in telecommunication, Compute Science OR equivalent
- Experience in J2EE, SOA, Spring, Hibernate, Struts, XML, REST, HTML5 development
- Experience in cloud computing development, such as SaaS, PaaS, etc.
- Strong programming skills, at least 3+ years experience in Java software development
- Familiar with the software development methodology
- Familiar with networking technology
- Good English communication skill
- Good team working spirit and communication skills, and hard working as well

Contact：hr@orange-ftgroup.com.cn

北京神舟航天软件技术有限公司

神舟软件是由中国航天科技集团公司控股的重点高新技术软件企业，公司业务领域涵盖工业软件与服务、电子政务与公共服务、基础软件、系统集成、航天文化创意等方面，拥有一批具有自主知识产权的实用型软件产品，产品与服务遍及机械制造、医疗卫生、教育等多个领域。

因发展需要诚征人才，我们会提供广阔的发展空间和完善的保障。

- 高级PLM业务架构师
- 高级Java系统架构师
- 高级Java开发工程师
- 项目经理/产品经理
- UI架构师
- 中级Java开发工程师

更多职位信息可浏览：http://www.bjsasc.com/rlzyavidm.html
简历投递：zhaomengyi@bjsasc.com　**联系人：赵梦祎**

无锡迈达博尔信息咨询有限公司

无锡国家软件园，占地800亩，建成55万平方米，在建37万平方米，入园企业超400家。
网址http://wnd.iparkhr.cn/

招聘职位：

- 无锡国家软件园——对日软件工程师
- 无锡国家软件园——Java开发工程师
- Android开发工程师
- 测试工程师

简历投递：iparkhr@126.co

方正国际软件有限公司

方正国际医疗卫生业务是中国医疗信息系统解决方案的提供商，经过16年的发展，已成为成熟的数字化医院及区域医疗卫生网解决方案的提供商和服务商。

为适应业务的蓬勃发展，现急聘：

Java中高级开发人员

投递邮箱：bfi_zhaopin1@founder.com
公司主页：www.founderinternational.com
公司地址：北京市海淀区北四环西路52号方正国际大厦5层

新书上架

本月新书



你必须知道的.NET

作者：王涛

来自微软MVP的最新技术心得和感悟，内容涉及.NET底层知识、.NET框架和CLR。除了探讨.NET的基本问题，作者对.NET的学习方法和学习工具均有所涉猎。全书以C#语言实现所有的代码示例，不可避免地分析了C#语言的某些特性，从而以更全面的角度阐述.NET 框架。



云计算宝典（技术与实践）

编著：虚拟化与云计算小组

以概述云计算统领全书，系统地探讨其在重要行业中的应用场景和方针，随后介绍了云计算的技术基础：虚拟化，并在此基础上定义了云架构，最后以勾勒云计算生态系统的动态和趋势作结尾。附录中列出了丰富的参考文献。



让Oracle跑得更快2：基于海量数据的数据库设计与优化

作者：谭怀远

主要讨论在海量数据的情况下，高效Oracle数据库的设计与优化。围绕此核心介绍了具体技术的应用，包括分区、索引的选择和使用、数据库对象的属性设置、并行技术、只读表空间、初始化参数的设置，几种常见的数据库架构和设计，以及在特定数据库架构下数据库的备份和恢复等技术。



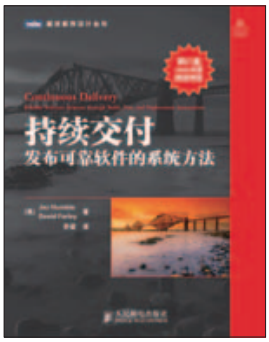
WCF服务编程（第三版）

作者：Juval L wy

译者：徐雷　徐扬

介绍WCF相关知识点，涉及WCF架构和基础构件，包括可靠性和传输会话等关键概念。从本书中，读者将学会如何使用服务托管、实例、并发管理、事务、无连接队列调用、安全和服务等内置特性。第三版更新内容涵盖了WCF 4.0新增加的功能和特性以及Azure AppFabric服务总线的内容。

推荐图书



持续交付：发布可靠软件的系统方法

作者：Jez Humble David Farley

译者：乔梁

出版社：人民邮电出版社

十年前，敏捷宣言发表，什么是敏捷，知者甚少。如今软件开发者能够开口即言敏捷，而无须担心没有听众：敏捷软件开发已成主流。作为敏捷开发最佳实践之一的持续集成（CI），近年来也被广泛接受。CI使得整个开发团队保持同步，消除了集成问题引起的延期，但CI只是第一步。软件即使被成功地集成到代码主干上，也仍旧是没有在生产环境中发挥作用的软件。这就是所谓的“最后一公里问题”，即如何让软件从“开发完成”迅速实现“上线发布”。

本书的两位作者在Thoughtworks工作期间，面临的项目有着共同的主题，那就是缩短从想法到可用软件之间的生产周期。尽管他们通常不会每天发布，但现在每两周发布一次却是很常见的，频繁且可靠地进行交付已然成为一种文化。他们和同事已经将很多每年才能做一次软件部署的组织带到了持续交付的世界里，让发布变成了常规活动。

有了上述经验积累，Jez和David阐述了“最后一公里问题”的解决办法，描述了如何构建部署流水线，才能将已集成的代码转变为已部署到生产环境中的软件。

这种交付思想长期以来一直是软件开发中被人遗忘的角落，是开发人员和运维团队之间的一个黑洞。因此毫无疑问的是，本书中的技术都依赖于这些团队的凝聚，而这也就是悄然兴起的DevOps运动的前兆。这个过程也包括测试人员，因为测试工作也是确保无差错发布的关键因素。贯穿一切的是高度自动化，让事情能够很快完成而且没有差错。

为了实现这些，需要付出努力，但所带来的好处是意义深远的。持续时间长且强度很高的发布将成为过去。软件客户能够看到一些想法快速变成他们可以每天使用的可工作软件。也许最重要的是，消除了软件开发中严重压力的一个重大来源。没有人喜欢为了让系统升级包能够在周一的黎明前发布而在周末紧张加班。

作者著书的目的在于告诉开发者如何做到无压力且频繁地交付软件：通过将敏捷和持续集成的理念应用到整个软件生命周期中，利用各种敏捷原则与最佳实践打破了用户、交付团队及运维团队之间的壁垒，让原本令人紧张、疲惫的软件发布过程变得轻松，令原本枯燥易错的部署操作已变得只需轻点鼠标即可完成。

还需注意到的是，这是首次从业务视角阐述了持续交付的必要性，并从现实问题出发，对软件交付过程进行了彻底的剖析，指出了各交付环节所需遵守的原则与最佳实践，列举了各种常见的反模式。案例翔实得当，贴近生产一线。

持续交付以全面的版本控制和全面自动化为核心，通过各角色的紧密合作，力图让每个发布都变成可靠且可重复的过程。最后值得一提的是，本书获得了2011年Jolt图书大奖，这也是对作者们辛勤贡献的巨大褒奖。



新机器的灵魂
作者：Tracy Kidder
译者：龚益　高宏志
出版社：机械工业出版社

本书曾经获得普利策奖。计算机业最振奋也是最残酷的特性就是喜新厌旧。无论是产品、技术，还是公司和人，最可怕的结局就是一不留神被时间的浪涛打翻。而这又是命中注定的唯一结局。似乎计算机业内对任何事物的历史指向只有一个：贬值，然后消失。

本书讲述了一群忘我工作的电脑技术人员制造一台新型计算机的过程。故事发生在20世纪70年代末，那时PC刚出萌芽，计算机市场正是小型机的天下。故事的背景还不是硅谷，而是麻省，主角则是已被历史波涛淘尽的通用数据公司（DGC）。

1966年，DEC将PDP—8投放市场一举成功，暴得大财，然而PDP—8的设计者爱德华·德·卡斯特罗却受到冷落。1968年4月，卡斯特罗等3位工程师负气出走，自行创业，创建了自己的公司——DGC，并于1969年率先推出了16位小型机系列产品，成功打破了DEC一统天下的局面。

1978年，创建仅10年的DGC闯进了《财富》工业企业500强之列。但此时，包括DEC在内的几家竞争对手推出了32位超级小型机（VAX等）。

面对突如其来的VAX该如何阻击，DGC的领导层并无准备，工程师汤姆·韦斯特提交了以“鹰”命名的机器，这也是未来VAX的真正杀手和克星。

为了知己知彼，韦斯特把和VAX有关的技术材料通读，甚至冒充DEC的员工混进VAX机房，将VAX的系统结构完全摸清。

就这样，DEC和DGC之间一场紧张、刺激、混杂着狂热的追逐战开始了。在经历了一系列无休止的“丛林战争”之后。“鹰”计算机大获成功。1980～1984年期间，DGC由于新机型及高级综合电子办公软件的引入，每年的毛利润大幅增长。

然而螳螂捕蝉，黄雀在后，一场发轫于1981年的PC革命已经兴起。在价格低廉的PC冲击下，1985年后DGC开始走下坡路，虽然能够默默无闻地经营着，但在IT历史上再无大的作为，当年曾经更加辉煌的对手DEC，在1998年被Compaq收购。不久Compaq又在2002年被惠普收购。2011年惠普宣布剥离个人电脑业务，高层动荡不断。

在这样快速的新陈代谢、疯狂的节奏下，一些内在的东西是弥久的，甚至是永恒的。毕竟任一行业，其真正的主角都是人，而人恐怕是世界上最难升级的事物。可以说，本书定义了计算机业整整一个时代的灵魂和精神。尽管业内技术2年一变，人物5年一换，但这个产业的问题和驱动力与10年、20年前相比并无两样，只是不同的公司扮演着不同的角色而已。无论是后来的PC之火，还是互联网浪潮，都没有脱离贯穿这个产业数十年的本质。

同样的故事不断重演，这就是IT业的美丽和动力所在，这些人前赴后继、推陈出新，使得产业活力不减。尤其是刚过去的互联网热潮、现如今的移动浪潮，更是以无数年轻人的“透支式生存”为代价。他们在行业里拼搏，以自己的青春为燃料，超速燃烧。

全球排行榜

Amazon

01 C Programming Language
02 Artificial Intelligence: A Modern Approach
03 Programming in Objective-C (3rd Edition)
04 Head First Java, 2nd Edition
05 iOS Programming: The Big Nerd Ranch Guide
06 Visualize This: The FlowingData Guide to Design, Visualization, and Statistics
07 JavaScript: The Definitive Guide: Activate Your Web Pages
08 Head First HTML with CSS & XHTML
09 JavaScript: The Good Parts
10 Effective Java (2nd Edition)

天珑书局（中国台湾）

01 前进Android Market！Google Android SDK实战演练
02 学徒模式：优秀软件开发者的养成之路
03 Eclipse完全攻略：从基础JAVA到PDE外挂开发
04 Google Android SDK开发范例大全
05 jQuery锦囊妙计
06 Google Android应用程序开发实战
07 Google！Android 3手机应用程序设计入门
08 ASP.NET 4.0专题实务—使用C#
09 一定要学会的HTML5＋CSS3网页设计实作应用
10 ASP.NET 4.0专题实务

第二书店

01 深入理解Android
02 Debug Hacks中文版：深入调试的技术和工具
03 Java编程思想
04 高可用MySQL：构建健壮的数据中心
05 浪潮之巅
06 Android内核剖析
07 松本行弘的程序世界
08 智能Web算法
09 大话处理器：处理器基础知识读本
10 面向模式的软件架构（第5卷）

GEEK产品



Power strip Svintus

数码设备越来越多，插座越来越不够用。而这只粉嫩小猪插座浑身布满了17个两孔的猪鼻子，每个猪鼻子都是独立的插孔，足以应付众多的的设备，且鼻孔相互之间保持合适的距离。不过如果是三相的插头恐怕就没办法了，因为猪只有两个鼻孔。



Chevrolet EN-V

通用雪佛兰在天津生态城引入最新的EN-V概念车。EN-V是一款由锂离子电池驱动的零排放车辆，使用标准的住宅电力，通过传统的插座充电，一次充电可行驶至少40公里，适应大多数市区旅程。EN-V通过结合GPS与车辆间通信和距离感应技术，可手动或自动驾驶。在自动驾驶模式下，EN-V可为不能操作车辆的人们代步。

Kodak PLAYFULL waterproof Ze2

柯达发布了一款口袋数码摄像机“PLAYFULL Waterproof”，仅重85g，可实现在水下3米处进行长达2小时的拍摄，还具备1.5米防摔和防尘能力。摄像机采用1/4英寸CMOS感光元件和等效焦距为31mm的定焦镜头，具有2倍数码变焦功能，可拍摄720p高清视频，还有HDMI输出接口、弹出式USB 2.0接口，一个SD卡槽，支持最大容量为32GB的SD/SDHC卡，可记录10小时内容。







2MP Mini Business Card Scanner

最小的名片扫描设备，只有41（长）mm×41（宽）mmX23（高）mm的大小。操作极其便捷，只需用USB连接到电脑上，将名片固定在展开后的前端，然后配套软件即可扫描、识别名片上的信息。此外，这个折叠式的小设备还可以作为一个网络摄像头使用。



MobiUS SP1

由Mobisante推出的MobiUS是世界上首个基于智能手机的超声波系统，通过USB连接，运行在Windows Mobile 6.5的Toshiba TG01上。这款小型医疗扫描设备配备有超声探针，可与移动设备和连接设备进行互联，具备WiFi和手机连接功能，可以在医疗点就地使用，价格比全尺寸的超声波设备更易接受，也更方便操作。可存储8GB的图像，图像可通过无线方式在线共享，也可以通过USB接口离线下载。不过其图幅较小，最大的分辨率也只是480×480像素，尚不能替代传统的超声波设备。





Ray Solar Charger

出门在外，再也不必担心没电。由Quirky推出的Ray太阳能充电器通过USB给手机或者其他设备充电。通过其内置的电池，可以在没有连接设备的时候储存备用。太阳能板上的吸盘结构支持多个位置放置，很容易地将其固定在窗户玻璃上。

Kandenchi

Elecom推出以电池为造型的无线鼠标Kandenchi。该鼠标的滚轮及鼠标按键置于侧边，上面是左键，下面是滚轮，可用食指、中指操控。右键在上方电池的正极部分，操作方式类似立式鼠标，轻巧易于携带。



商务才是关键点

专访6688.com创始人王峻涛

记者 / 陈秋歌

王峻涛，中国电子商务早期开拓者，其网名和笔名“老榕”早已广为人知，IT技术专业出身，从事近10年技术研发工作后转身投入电子商务行业，开启了国内电子商务的新大门，是中国第一批互联网人物。

自学成才，计算机系里的文学爱好者

《程序员》：请简单谈谈您的童年经历。您在大学生活中印象最深刻的事情是什么？

王峻涛：我对童年的记忆特别深刻。我仅用四年时间完成了五年的小学课程。利用初中的时间自学了高中课程，在上高一不久就参加了高考，并取得了福建20岁以下组的第一名，被哈尔滨工业大学计算机专业录取。上大学时我才15岁。



王峻涛自1999年创办8848网站以来，一直在电子商务行业里耕耘，是中国B2C电子商务的早期探索者

报考计算机专业主要是受叶永烈编写的《电子计算机》一书的影响。1978年，国内图书匮乏，尤其是科普类图书，叶永烈当时以写科普图书为主。读完这本书，我就期盼自己能在大学时学习该专业。哈尔滨工业大学在计算机软件专业上的课程设置和现在差不多，甚至包含了现在的一些硕士课程。但当时国内条件还比较差，没有PC机，我们编写完程序，首先需要把它转化成十六进制，并在打孔机上打孔。即便一个很简单的计算题目，也需要耗费大量的纸，一旦出错，就会造成纸张的浪费。当时使用的计算机比较庞大，调试机会也很少，所以当时我们编程不仅要保证程序的正确性，还要保证程序的可检查性。

业余时间我还十分喜欢写诗、写小说。1978年国内刚恢复稿费制度。靠稿费收入，我在大学的生活完全可以自己自足。关于我的写作能力，高考时还有一件趣事。高考阅卷老师为我的作文评了100分，但似乎意犹未尽，破例又多给了1分，最后我的作文以101分成为福建考生中高考作文最高分。

从业经历，激发自主创业

《程序员》：1982年大学毕业后，您从事的第一份工作是什么？当时从业经历如何？

王峻涛：大学毕业后，我顺利考取了研究生。当时正好有一家北京军工类企业来招聘，我便放弃了在哈尔滨工业大学继续读研的机会，应聘来到了北京。

我们属于恢复高考后的第一批大学毕业生，当时全国各地程序员奇缺，各个单位的程序员大都是老同志，多靠自学转行做程序员。入职后我便迅速成为研发部门主力，不到半年，因工作突出被提拔为系统分析员，不久又被提升为项目主管。四五年后，我便开始做项目管理，成了一位管理者。

《程序员》：您曾有过一段海外工作的经历，这段经历对您有什么影响？

王峻涛：在20世纪80年代末，因工作的需要，我去了美国，印象最深的是美国的创业环境。

当时，我遇到了一位研究发动机的专家，他从美国军工企业退休后，开始在自家车库中创业。他利用自己掌握的技术及部分专利，研究改良军用发动机并大大降低成本，以便民用。在他专门改良其中的电脑控制系统时，我总会利用闲暇时间，帮他调试程序。

当他把原理性样品展示给福特汽车公司后，福特马上投资了三百万美元，资助他对该产品进行工业化改进。他在该产品的基础上发明了多点燃油喷射电子系统，得到了福特的认可和大规模使用。这也促使了该创业公司迅猛快速地发展，最终被福特以巨额收购。可以说现在福特在多点喷射方面的领先技术与这位专家有紧密的联系。

美国的这段经历，不仅让我体验到了创业的过程，同时也让我了解到美国自由的环境推动了其技术的发展。在美国，有很多像他这样的人，他们以创新的思维，自由地创业，风险投资对他们的发展也起到了非常大的作用。我后来创业，在很大程度上也和这段经历有关。

《程序员》：您首次创业是什么时候？创业过程如何？

王峻涛：从美国回来后，国内正赶上百万大裁军，很多军工企业开始转为民营企业。当时国内也正进行改革开放，为了寻求更有挑战性的工作环境，我主动申请转到了民营企业，三年后，我回到福建创办了一家IT公司。

当时是1991年，我创办的这家公司一方面代理其他公司的软件，如用友财务软件及当时很重要的汉字平台等；另一方面帮助中小企业的IT系统做系统集成，如果企业有什么特殊需要，我们还会帮他们做二次开发。

公司创办之初，仅有十几个人。依靠多年国有企业的工作经历及个人在家乡的名气，很受当地政府和企业的信任，所以这次创业过程还是十分顺利的。

1995年，我们加盟联邦软件销售连锁组织开始卖套装软件。其中有一款软件产品，为我们赢得了大笔收益。在一次去大学做客座老师的偶然机会，我发现有个学生编写了一个练习五笔字型的小软件，与当时难学的五笔字型相比，它轻便易学，十分不错。我买下了这个软件并做了改造，商业包装后，取名为“轻松学会五笔字型”，让我没想到的是，这款软件后来竟然流行了近20年，并与一些PC进行捆绑。

投身互联网，开启电子商务新大门

《程序员》：您什么时候开始关注电子商务？当时是如何开拓国内电子商务市场的？

王峻涛：从1995年我就开始关注互联网，直到1999年，联邦软件成立了电子商务部，我便来到北京做8848网站。从此我与软件行业的接触慢慢减少，并逐渐成为一名管理者。

8848网站刚成立时仅有四个人，但后来8848却对国内电子商务的发展有着很重要的启蒙和引导作用，它告诉了大家什么是电子商务、电子商务该如何做、为国内网上购物流程确立了基本框架。8848后期的发展也相当顺利，唯一的困扰就是当时互联网用户少，才200多万人。而当时我们已经拥有了40多万用户。

2002年我离开了8848，并于同年8月，成立了6688网站。该网站成立之初，整个互联网和电子商务正处于最低谷时期。我们当时选择的主要业务是为客户提供电子商务解决方案。鉴于有的企业对电子商务系统的运营并不熟悉，我们同时还会为他们提供运营服务，包括运营管理、客服、发货等一系列供应链问题。其中北京琉璃厂文化园区是我们主要运营的一个项目，我们负责该园区的所有互联网事务。

《程序员》：您觉得做好电子商务的关键是什么？

王峻涛：商务，而非技术。第一，要明确你的供应链，你有什么东西可卖，如何卖得过别人；第二就是学会做服务，通过互联网做服务，与地面上做服务很不同；第三是管理，一定要对投入合理控制，要学会控制成本。一旦成本升高，就很难再降下来，而很多电子商务企业，虽然越做越大，但越做越困难，问题可能就出于此。

在技术上，电子商务企业应该更多了解一些技术趋势。即便你不是技术出身，也要从外行的角度有部分了解，至少你要知道自己需要什么，然后去挑选适合自己的电子商务平台。

《程序员》：对于当前从事IT技术的研发人员，您有什么建议？

王峻涛：建议现在的程序员踏踏实实做好编程工作，认认真真地写好每一行代码。今后无论你想去经商或者继续在软件行业中更深地钻研下去，这一点十分重要。其次就是要站在用户的角度去开发软件。无论程序结构多好、可靠性多高，它如果不能满足用户的需求，终会被用户遗弃。

对于开发者，还要多关注行业发展趋势，不要死守自己已经掌握的技术。同时也要对国际和国内行情有一定了解，具有一定的前瞻性。

Mac OS X 背后的故事（六）

上善若水

Cordell Ratzlaff 引发的 Aqua 革命（下）

文 / 王越

上期讲到，Cordell Ratzlaff新界面方案得到Steve Jobs的高度肯定，Steve Jobs让各开发组紧紧围绕在界面设计组周围，共同建造Mac OS X。此时，离Mac OS X第一个公共测试版的发布，仅有一年半时间。这时苹果的设计构想，还仅仅是个概念，在本篇中我们将展示Aqua的具体设计过程。

设计与软件的融合

开发分设计和软件两条路并行走，“两手抓，两手都要硬”。

设计是个有趣的领域。有些人认为，设计就是产品的外观看上去什么样。但其实，如果细想一下，你会发现设计其实是有关产品如何工作的学问。

——Steve Jobs

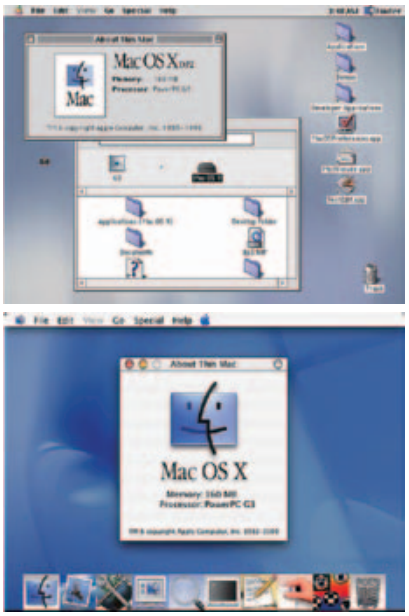
首先，苹果定下计划，并规划整个界面设计元素的方案，把设想通过可操作性强的材料让工程师来实现。

Cordell Ratzlaff每周都要和Steve Jobs开会，向他展示界面设计小组最新成果。任何大家现在见到的各界面控件，如菜单、按钮、进度条、Steve Jobs都一一过目，毫不马虎。针对每一个控件，Cordell Ratzlaff会要求拿出多套方案来，让Steve Jobs选出他中意的。Steve Jobs也会提出各种他自己的见解和改进建议，而Cordell Ratzlaff则会根据这些回馈不断修改，直到Steve Jobs满意为止。

与此同时，软件工程师也以越来越重的比例加入到这个设计行列中。

图形界面设计小组使用的设计软件是Macromedia Director。它能做出演示用的动画，可以演示打开、关闭窗口、下拉菜单等模拟效果，但这些并不是可供用户使用的最终软件。软件工程师需要把图形界面设计师的设计，变为一行行代码，运用到Mac OS X中。所以每次会议的Macromedia Director动画演示机旁，还会有一台计算机，预装了软件工程师转换的代码。当工程师们向Steve Jobs展示最新代码如何工作时，Steve Jobs会身体前倾，鼻子快贴到荧幕上，观察细微到“像素级别”来比较软件的表现和之前的设计是否完全一致。如果他有发现任何细微的差错，一阵类似“你们全是一帮白痴”的腥风血雨就会在办公室中展开。

设计整套方案是一个令人难以置信的漫长过程，尤其是遇到追求完美的Steve Jobs。Mac OS X中有一个控件叫滚动条（NSScroller）。当需要显示的内容长于当前控件大小时就会出现滚动条，可上下翻阅内容。这是一个非常不起眼的控件，大多数时间，用户甚至注意不到它的存在，甚至在十年后的今天它都被默认不显示了（关于Lion图形界面的改动受iOS思潮的影响我们今后会提到）。但哪怕是这种不起眼的细节，Steve Jobs都偏执地当个大项目来做。Mac OS X的界面设计是有史以来最复杂的一个，需要考虑诸多因素——比如所在窗口的活动与否，都会影响这个控件的颜色等属性。就滚动条而言，箭头的大小、位置的变化、颜色的启用等全都是活动的属性，牵一发而动全身。一根看似简单得不能再简单的滚动条，设计组花了整整六个月来修改。



Developer Preview 从2到3的变化也是Aqua推进的过程

当时，Mac OS X的用户界面有两个重大的设计目标：第一是让老用户没有压力地迁移过来，且倍感新界面的好用；第二是让那些从未摸过Mac的人尽快上手，并称赞这界面很好很强大。所以，整个界面设计保留了老Mac OS界面元素的设计理念，但同时又对很多有问题的老设计进行了革新。比如，在老版Mac OS中，各种系统设置选项是隐藏在不计其数的系统扩展、控制面板，以及很多系统组件中的。用户要想联个网，要去五六个地方设网络、设IP、设连接设密码，而在Mac OS X中，所有这些设置都被分门别类地规类到一个单一的程序——系统首选项（System Preferences），让用户“足不出户”，就能进行一切相关设置。【注：参见链接http://www.cultofmac.com/how-mac-os-x-came-to-be-exclusive-10th-anniversary-story/87889】

精简的狂热追求和大胆的设计创新

Apple偏爱最简化的设计，而往往满屏的窗口让Steve Jobs忍无可忍。又酷又炫的Dock横空出世，巧妙地解决了这个问题。Dock的设计源于Mac OS X的前身NeXTSTEP，但在Mac OS X中完全被重写，并重定义了它的功能。Dock提供用户一个放置常用软件图标、闲置窗口、文档的场所，Steve Jobs说“任何东西都能被拉进 Dock”。但Dock真正神奇的，是它犹如多拉A梦的口袋，有无限的承载能力。当放入Dock中的东西变多时，它会自动把横向宽度变长、图标变小，可承载几十个窗口。当窗口缩入和还原时，都配有“精灵”一样的动画——在Dock的图标多的时候，每个图标很小，用户就很难找到需要的——灵动且放大动画可以让用户能快速地找到所需。

另外，起初版本的Dock中每个图标都是正方形的方块，被换成半透明的背景，看得人垂涎欲滴。这些经典的设计，影响了整整一代图形界面设计者，被各山寨界面抄了一遍又一遍，甚至又活在当今的Ubuntu Linux的Unity和Windows 7中。

Apple追求清爽甚至到了发疯的地步，在最初版的 Mac OS X Public Beta中，每个窗口有一个按钮，只要按下，除了当前窗口外，其它一切都会飞入Dock。因此，只要一键，“整个世界都清静了”。而在后来每个版本的Mac OS X中，都有大的更新来防止窗口或其他界面元素的堆积。10.4时代的Expose【注：参见链接http://en.wikipedia.org/wiki/Exposé_(Mac_OS_X)】，10.5时代的Stack【注：参见链接http://en.wikipedia.org/wiki/Stacks_(software)】和Spaces【注：参见链接http://en.wikipedia.org/wiki/Spaces_(software)】，10.6时代的Expose和Dock相结合双管齐下，到 10.7时代的Mission Control，都是用来解决果面精简这一个问题的。

而很多传统的界面控件也被赋予了新的含义。比如 Steve Jobs觉得，“最大化”一个窗口没有实际意义，而且把整个窗口最大化，也会挡住后面的窗口（直到2011年，Apple用“全屏”来重新定义传统的“最大化”）。而Mac OS X没有所谓的“最大化”，取而代之的是自动计算后调整窗口到所需大小的“最适化按钮”。而关闭一个窗口的含意也不该是关闭一个程序，而只应是结束目前的内容。Apple的许多设计都格外具有魄力，完全重写了界面设计的教科书。当然，有许多地方Apple确实做得矫枉过正，比如Apple一直是我见过的只有拖住右下角才能改动窗口大小的唯一系统。这个置用户于不顾的狂妄设计，一直在十年后发布的 Lion中，才得以改变。

Steve Jobs一直是界面设计的重要顾问。他有时候会提出一些看似稀奇古怪的意见，但往往最终又被证明是好的。比如，有一次他在会上指出，窗口左上角的“关闭”、“最小化”、“最适化”三个按钮的颜色都是一样的灰色，不容易区分他们。他建议把三个按钮变成交通灯的颜色，并且当鼠标移到附近时，显示出相应的图形指示。当Cordell Ratzlaff一群人听到这个主意后面色大变，认为简直是计算机图形设计史上最好笑的段子——谁会把电脑当交通灯使啊。不过改完后，他们对Steve Jobs心悦诚服——“红灯给用户一个终止的警示，这个窗口要被关掉；黄灯表示这个窗口要被放入等待队列，以便以后再通行；最适化则是给这个窗口大开绿灯”——这样高明的比喻，使Cordell Ratzlaff对Steve Jobs崇拜得五体投地。

18个月转瞬即逝，“你们就是一群白痴”的骂声依旧清晰，而此时的Mac OS X的图形界面，已今非昔彼。

“语静声息。我走上舞台。依着那打开的门，我试图探测回声中，蕴涵着什么样的未来。”（北岛翻译的帕斯捷尔纳克的《哈姆雷特》）。

18个月后的2000年1月，新世纪的钟声刚刚敲响，Steve Jobs镇定地走上MacWorld大会的舞台，独领风骚的新世纪的经典大作Aqua，此时，就要被他揭开帷幕。

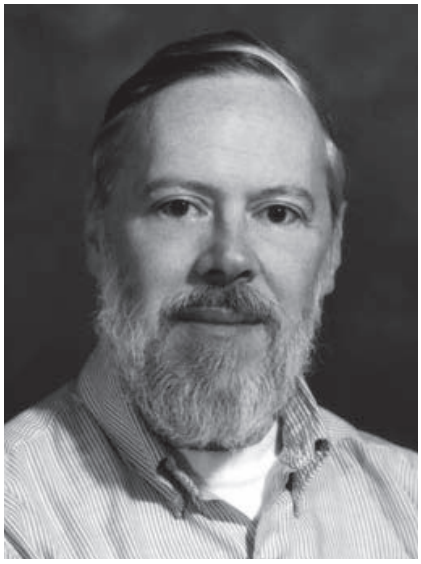


王越

清华大学建筑学院毕业，现居美国。中国著名TeX开发者，非著名OpenFOAM开发者。

责任编辑：白羽中（baiyz@csdn.net）

软件传奇
Dennis M. Ritchie



文 / 佳琦

Dennis MacAlistair Ritchie，C语言之父，1941年9月9日出生于美国纽约。在技术圈里，他常被称为DMR（这是他在贝尔实验室的用户名，后来成为他常用的代号）。

DMR曾在哈佛大学研习物理和应用数学，1967年进入贝尔实验室，并任朗讯系统软件研究部门领导人。1983年他和Kenneth Lane Thompson 一同获得了图灵奖，以表彰他们对“研究发展了通用的操作系统理论，尤其是实现了Unix操作系统”的贡献。1999年两人又因创造发展C语言和Unix操作系统获得了美国国家技术奖章。

DMR的父亲Alistair E. Ritchie是他的第一位人生导师。Alistair Ritchie长期担任贝尔实验室科学家一职，在电路晶体管理论方面颇有造诣。在父亲的影响下，DMR在大学开始对计算机着迷，那时的计算机还是古老的打孔卡片设备。DMR更加着迷于计算机处理的理论和实际问题，1968年DMR获得数学博士学位的论文，正是计算机理论相关的《递归函数的层次》。

加入贝尔实验室不久，DMR就参与了Multics项目，负责多道处理机的BCPL语言和GE650的编译器，它们都属于GECOS系统。同样的，他也写了ALTRAN语言的代数编译器，那是用于符号计算机的一种语言和系统。

在工作中，DMR遇到了对他职业生涯影响最大的人——Ken Thompson。DMR曾表示Unix大部分是Ken的工作。不同于DMR对理论的偏好，Ken是一位电子发烧友，与DMR形成互补。当时贝尔实验室在Multics上遭遇挫折，于是两人在一台小型机上重新开发了这个系统，并将其命名为Unix。DMR为Ken提供了扎实的理论支持，三年后，Unix诞生。

Unix风靡之前，将其移植到其他机器上总是会遇到麻烦，DMR和Ken决定采用更为先进的程序语言来重新编写Unix，研发的任务交给了DMR。

于是，DMR向Ken的B语言中加入数据类型和新的语法，创造了一种新语言——C。C语言也成为Unix的基础，在其他领域得到广泛应用：从应用程序到操作系统，从移动设备到超级计算机，都能看到C语言的应用。Stroustrup编写C++语言时，也曾以C语言作为参考。

1973年，重新编写的Unix上线，成为现代通用性操作系统。此后，DMR与Steve Johoson合作成功地将这个操作系统移植到Interdata 8/32上，这种方式也证明了Unix的可移植性，从此Unix快速传播开来。这次移植还有更重大的意义：摆脱硬件束缚的开放系统由此诞生了。

完成了Unix和C语言之后，DMR并未停止创新工作，他继续发展了两款软件，包括1995年发布的Plan 9操作系统和1996年发布的Inferno操作系统以及Limbo语言。两者均是基于他以前的工作上发展的。1978年DMR和Brian W.Kernighian合作出版的《The C Programming Language》，成为C语言方面最权威的教材，也是后来所有语言著作的范本。

如今，人们评价C语言时，总是会说，它是现代程序语言革命的起点，是最有效、最通用的编程语言，是使用最广泛的编程语言之一。当人们评价Unix操作系统时，也总是会说，它是历久弥新的操作系统，是信息社会的基石，不但是一种技术，更是一种文化。然而，它们的发明者DMR却保持一贯的谦逊和低调，在贝尔实验室的研究部，从事自己喜欢的研究工作，著书立说，潜心编程，对于自己取得的成就也从不傲功。曾有人评价Unix是继晶体管以后第二个最重要发明，DMR回应这种说法时言道：“就其受公众瞩目程度而言，Unix的确是一项重要的发明。”他认为C语言虽然十分成功，但迟早会被其他语言所取代。

2011年10月9日，DMR故于伯克利的家中，享年70岁。虽然他离世的消息在非IT领域并未引起太多的注意，但他取得的光辉成就在计算机的历史上无法比拟。我们赖以运行其上的信息社会的基础，有他一份重要的奠基。



0
最近我研究了一下写文档的方法论，希望大家都能按照新的标准来做，把项目文档规范起来。

1
天哪，他居然雇了个外部的方法学专家拟了一套标准模板。
没关系，我们可以在项目组之外雇一名实习生抄写员来干这个。

2
文档写完了吗？让我来核查一下
实习生（抄写员）

3
你是用编辑器写的吗？
我用了Word。
你用了布局、全局格式、字体和系统编号吗？
是的。全用了

4
文档中包括术语词汇表、业务规则、参考文献和背景分析吗？
是的，都有。

5
模板的每个标题下面都有写上足够的字吗？
是的，我保证。

6
哦，太好了。现在我们可以让领导签字同意了。

7
方框里写字了吗
是的
这个方框呢？
同上



作者：
西乔
设计师，项目经理。06年起携创业团队从事Web技术外包开发及产品咨询顾问。
如果你有什么好玩的关于程序员的故事、对话、代码，愿意通过漫画的形式分享，请给西乔发邮件：arthur369@gmail.com

